



พาร์มลี เนเบอร์

โดย

นายณรงค์ ทองอุ่ยง  
นายศิวกร เกิดลาภี

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
วิทยาศาสตรบัณฑิต  
สาขาวิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์  
ปีการศึกษา 2567  
ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

ฟาร์มลี เนเบอร์

โดย

นายณธรรມ ทองอุ่ยง  
นายศิวกร เกิดลาภี

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
วิทยาศาสตรบัณฑิต  
สาขาวิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์  
ปีการศึกษา 2567  
ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

Farmly Neighbour

BY

Mr.Natum Tongiyang

Mr.Siwakorn Kerdlaphee

A FINAL-YEAR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE  
COMPUTER SCIENCE  
FACULTY OF SCIENCE AND TECHNOLOGY  
THAMMASAT UNIVERSITY  
ACADEMIC YEAR 2024  
COPYRIGHT OF THAMMASAT UNIVERSITY

มหาวิทยาลัยธรรมศาสตร์  
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

นายณัฐรัม ทองเอียง  
นายนิกร เกิดลาภี

เรื่อง

พาร์มลี่ เนเบอร์

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์  
เมื่อ วันที่ 30 พฤษภาคม พ.ศ. 2568

อาจารย์ที่ปรึกษา

น.ส. ฤกษ์ ชลารักษ์

(อ.ดร.นวลดรุษ ชลารักษ์)

กรรมการสอบโครงการพิเศษ

ผศ.ดร.กฤษติศ ชาญเชี่ยว

กรรมการสอบโครงการพิเศษ

ก.ก.ก. ก.

(รศ.ดร.ธนาธร ทะนานทอง)

มหาวิทยาลัยธรรมศาสตร์  
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

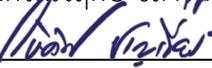
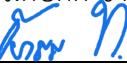
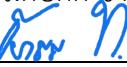
นายณัฐรัม ทองอุ่ย  
นายนิกร เกิดลาภี

เรื่อง

พาร์มลี่ เนเบอร์

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์  
เมื่อ วันที่ 30 พฤษภาคม พ.ศ. 2568

อาจารย์ที่ปรึกษา  
กรรมการสอบโครงการพิเศษ<sup>1</sup>  
กรรมการสอบโครงการพิเศษ<sup>2</sup>

\_\_\_\_\_  
(อ.ดร.นวฤกษ์ ชลารักษ์)  
  
\_\_\_\_\_  
(ผศ.ดร.กฤษดิศ ชาญเชีย)  
  
\_\_\_\_\_  
(รศ.ดร.ธนกร ธนาท้อง)  


(1)

หัวข้อโครงการพิเศษ

ฟาร์มลี เนเบอร์

ชื่อผู้เขียน

นายณธรรມ ทองເອີ້ງ

ชื่อผู้เขียน

นายศิวกร ເກີດລາກີ

ชื่อปริญญา

วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

สาขาวิชา/คณะ/มหาวิทยาลัย

สาขาวิชาวิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงการพิเศษ

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยธรรมศาสตร์

อ.ดร.นวฤกษ์ ฉลาดรักษ์

ปีการศึกษา

2567

## บทคัดย่อ

โครงการนี้มีวัตถุประสงค์เพื่อพัฒนาเว็บไซต์ e-commerce สำหรับชื่อขายสินค้าเกษตร โดยมุ่งเน้นให้เกษตรกรสามารถนำสินค้าของตนเองมาเผยแพร่และจำหน่ายได้โดยตรงถึงผู้บริโภคผ่านช่องทางออนไลน์ระบบที่พัฒนาขึ้นช่วยลดขั้นตอนการขายแบบตั้งเดิมเพิ่มโอกาสทางการตลาดและอำนวยความสะดวกแก่ทั้งผู้ซื้อและผู้ขายเว็บไซต์ถูกออกแบบให้มีฟังก์ชันหลักที่จำเป็นสำหรับระบบ e-commerce ครบถ้วนอาทิ การลงทะเบียนผู้ใช้งาน การจัดการสินค้า การสั่งซื้อ การชำระเงิน การรีวิวสินค้า รวมถึงระบบจัดการร้านค้า

ด้านเทคนิค เว็บไซต์ถูกพัฒนาด้วยเฟรมเวิร์ก Next.js โดยใช้ภาษา TypeScript และเชื่อมต่อกับฐานข้อมูล PostgreSQL ผ่าน Prisma ORM รองรับการเข้าสู่ระบบด้วย OAuth เช่น Google หรือ Facebook และมีการนำระบบขึ้นใช้งานจริงผ่านบริการของ Amazon Web Services (AWS) เช่น EC2, RDS และ S3 เพื่อให้สามารถเข้าถึงได้ตลอด 24 ชั่วโมง โครงการนี้แสดงให้เห็นถึงความเป็นไปได้ของการประยุกต์ใช้เทคโนโลยีเว็บแอปพลิเคชันเพื่อส่งเสริมภาคเกษตรกรรมไทยให้ก้าวสู่ระบบการค้าออนไลน์อย่างมีประสิทธิภาพและยั่งยืน

**คำสำคัญ:** e-commerce, เฟรมเวิร์ก, พัฒนาเว็บไซต์, เกษตรกร, สินค้าเกษตร

## กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณอาจารย์ที่ปรึกษาโครงงาน อ.ดร.นวฤกษ์ ชลารักษ์ ที่ได้ให้คำแนะนำ ความรู้ และข้อเสนอแนะอย่างต่อเนื่องตลอดระยะเวลาการดำเนินโครงงาน ซึ่งเป็นแนวทางสำคัญที่ช่วยให้ ข้าพเจ้าสามารถพัฒนาเว็บไซต์ e-commerce สำหรับสินค้าเกษตรนี้ได้อย่างสำเร็จลุล่วง

ขอขอบคุณคณะอาจารย์ในภาควิชา วิทยาการคอมพิวเตอร์ คณะ วิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ที่ได้ถ่ายทอดความรู้และสนับสนุนในด้านวิชาการตลอดหลักสูตรการศึกษา ทำให้ข้าพเจ้ามีพื้นฐานความรู้ที่จำเป็นต่อการพัฒนาโครงงานฉบับนี้

ขอขอบคุณเพื่อนร่วมทีมและเพื่อนนักศึกษาทุกคนที่ให้กำลังใจ และเปลี่ยนความคิดเห็น และช่วยเหลือกันในช่วงที่ดำเนินงานโครงงาน ซึ่งมีส่วนสำคัญต่อความสำเร็จของโครงงานนี้เข่นกัน

สุดท้ายนี้ ข้าพเจ้าขอขอบคุณครอบครัวที่ให้การสนับสนุนและกำลังใจอย่างเต็มที่มาโดยตลอด ทั้งใน ด้านเวลา ทรัพยากร และแรงใจ ซึ่งมีค่ายิ่งต่อการศึกษาและการทำงานในครั้งนี้

ขอกราบขอบพระคุณทุกท่านมา ณ โอกาสนี้ด้วยความซาบซึ้งอย่างยิ่ง

นายณธรรມ ทองอุ่ย  
นายศิวกร เกิดลาภ

## สารบัญ

	หน้า
บทคัดย่อ	2
กิตติกรรมประกาศ	3
สารบัญ	4
สารบัญตาราง	8
สารบัญภาพ	9
รายการสัญลักษณ์และคำย่อ	12
 บทที่ 1 บหนា	 1
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ของโครงการ	2
1.5 ข้อจำกัดของโครงการ	2
 บทที่ 2 วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	 3
2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง	3
2.1.1 การตัดสินใจของผู้บริโภคต่อการซื้อสินค้าเกษตรออนไลน์	3
2.1.2 ชาวสวนไม่มีความมั่นคงในอาชีพ	3
2.1.3 ตลาดเกษตรออนไลน์ยังไม่มีแนวโน้มเติบโตสูงขึ้น	5
2.1.4 ปัจจัยที่มีผลต่อการตัดสินใจเลือกซื้อผักและผลไม้สดของผู้บริโภคในระบบสายโซ่ความเย็น(Cold Chain)	6

2.1.5 มาตรฐานสินค้าเกษตรสำคัญอย่างไร และมีอะไรบ้าง	7
2.2 งานวิจัยที่เกี่ยวข้อง	8
2.2.1 DGT Farm	8
2.2.2 JatujakMall	9
2.3 เทคโนโลยีที่เกี่ยวข้องกับโครงการ	10
2.3.1 HTML (HyperText Markup Language)	10
2.3.2 CSS (Cascading Style Sheets)	11
2.3.3 Next.js Framework	11
2.3.4 PostgreSQL	12
2.3.5 Prisma ORM (Object Relational Mapping)	13
2.3.6 GitHub	13
2.3.7 TailwindCSS	14
2.3.8 supabase	14
บทที่ 3 วิธีการวิจัย	15
3.1 ภาพรวมของโครงการ	15
3.1.1 สถาปัตยกรรมของระบบ	16
3.2 การวิเคราะห์ขอบเขตและความต้องการของระบบ	18
3.2.1 วิเคราะห์ความต้องการของระบบในด้านของการทำงานด้วยตนเอง	18
3.2.2 วิเคราะห์ความต้องการของระบบในด้านของผู้ใช้งาน	18
3.3 การดำเนินงาน	27
3.3.1 วิเคราะห์ความต้องการ	27
3.3.2 ออกแบบระบบ	28
3.3.3 พัฒนาระบบ	39

3.3.4 ทดสอบระบบ	45
3.4 ประเด็นที่น่าสนใจและสิ่งที่ท้าทาย	45
3.4.1 ประเด็นที่น่าสนใจ	45
3.4.2 สิ่งที่ท้าทาย	46
3.5 ผลลัพธ์ที่คาดหวัง	47
3.5.1 ความสะดวกและความเร็วในการซื้อขายสินค้าเกษตรออนไลน์	47
3.5.2 การสร้างความเชื่อมั่นให้กับเกษตรกรและผู้บริโภค	47
3.5.3 การเพิ่มการเข้าถึงตลาดสำหรับเกษตรกร	48
บทที่ 4 ผลการดำเนินงาน	49
4.1 หน้าจอหลักของระบบ	49
4.1.1 UC01: Log in – Log out	49
4.1.2 UC02: Switch Role	51
4.1.3 UC03: Review – Comment	52
4.1.4 UC04: Check Buy History	53
4.1.5 UC05: Add to Basket	54
4.1.6 UC06: Check Out Product in Basket	55
4.1.7 UC07: Search Item	56
	56
4.1.8 UC08: Check Farmer History	57
4.1.9 UC09: Add Product (Agriculture)	58
4.1.10 UC10: Community Farmer	59
4.2 การทดสอบระบบเบื้องต้น	60
บทที่ 5 สรุป	63

5.1	ความสมบูรณ์ของฟังก์ชัน	63
5.2	เทคโนโลยีและเครื่องมือที่ใช้	64
5.3	ความสอดคล้องกับเป้าหมายโครงการ	64
	รายการอ้างอิง	65

## สารบัญตาราง

หน้า

ตารางที่ 3.2.2 กรณีใช้งาน (Use Case) ต่าง ๆ	20
ตารางที่ 3.2.2.1 กรณีใช้งาน (Use Case) Log in - Log out	22
ตารางที่ 3.2.2.2 กรณีใช้งาน (Use Case) Switch role	22
ตารางที่ 3.2.2.3 กรณีใช้งาน (Use Case) Review – Comment	23
ตารางที่ 3.2.2.4 กรณีใช้งาน (Use Case) Check buy history	23
ตารางที่ 3.2.2.5 กรณีใช้งาน (Use Case) Add to basket	24
ตารางที่ 3.2.2.6 กรณีใช้งาน (Use Case) Check out product in basket	24
ตารางที่ 3.2.2.7 กรณีใช้งาน (Use Case) Search item	25
ตารางที่ 3.2.2.8 กรณีใช้งาน (Use Case) Check farmer history	25
ตารางที่ 3.2.2.9 กรณีใช้งาน (Use Case) Add product (Agriculture)	26
ตารางที่ 3.2.2.10 กรณีใช้งาน (Use Case) Community Farmer	26
ตารางที่ 4.2 ผลการทดสอบระบบเบื้องต้น	62

## สารบัญภาพ

	หน้า
ภาพที่ 2.1.2 แสดงแผนภาพแสดงผลโอลอนภาคต์ขาวส่วนผลไม้ความสามารถในการแข่งขันและความมั่นคงทางอาชีพ	4
ภาพที่ 2.1.3 แสดงแผนภาพแสดงผลโอลโควิด-19 กับตลาดสินค้าเกษตรยุคดิจิทัล	5
ภาพที่ 2.1.5 แสดงสัญลักษณ์ของ WTO	7
ภาพที่ 2.2.1 แสดงหน้าตาของเว็บไซต์ DGT Farm	8
ภาพที่ 2.2.2 แสดงหน้าตาของเว็บไซต์ JatujakMall	9
ภาพที่ 2.3.1 แสดงสัญลักษณ์ของ HTML	10
ภาพที่ 2.3.2 แสดงสัญลักษณ์ของ CSS	11
ภาพที่ 2.3.3 แสดงสัญลักษณ์ของ Next.js	11
ภาพที่ 2.3.4 แสดงสัญลักษณ์ของ PostgreSQL	12
ภาพที่ 2.3.5 แสดงสัญลักษณ์ของ Prisma ORM	13
ภาพที่ 2.3.6 แสดงสัญลักษณ์ของ Github	13
ภาพที่ 2.3.7 แสดงสัญลักษณ์ของ TailwindCSS	14
ภาพที่ 2.3.8 แสดงสัญลักษณ์ของ supabase	14
ภาพที่ 3.1.1 แสดงสถาปัตยกรรมของระบบ	16
ภาพที่ 3.2.1 แสดงความต้องการของระบบในด้านของการทำงานด้วยตนเอง	18
ภาพที่ 3.2.2 แสดงความต้องการของระบบในด้านของผู้ใช้งาน	18
ภาพที่ 3.3.2 แสดงการออกแบบ UX/UI ของระบบ	28
ภาพที่ 3.3.2.1 แสดง Er diagram ของระบบ	29

ภาพที่ 3.3.2.2.1 แสดงถึงการทำงาน Log in - Log out	34
ภาพที่ 3.3.2.2.2 แสดงถึงการทำงาน Switch Role	34
ภาพที่ 3.3.2.2.3 แสดงถึงการทำงาน Review – Comment	35
ภาพที่ 3.3.2.2.4 แสดงถึงการทำงาน Check Buy History	35
ภาพที่ 3.3.2.2.5 แสดงถึงการทำงาน Add to Basket	36
ภาพที่ 3.3.2.2.6 แสดงถึงการทำงาน Check Out Product in Basket	36
ภาพที่ 3.3.2.2.7 แสดงถึงการทำงาน Search Item	37
ภาพที่ 3.3.2.2.8 แสดงถึงการทำงาน Check Farmer History	37
ภาพที่ 3.3.2.2.9 แสดงถึงการทำงาน Add Product (Agriculture)	38
ภาพที่ 3.3.2.2.10 แสดงถึงการทำงาน Community Farmer	38
ภาพที่ 3.3.3.1 แสดงถึงไฟล์ .env	40
ภาพที่ 3.3.3.4.1 ตัวอย่างการเอา Component มาใช้งานโดย ส่ง id ที่เป็นของ product ไปให้ด้วย	42
ภาพที่ 3.3.3.4.2 ตัวอย่าง Component ที่ถูกเรียกไปใช้	42
ภาพที่ 3.3.3.5 ตัวอย่าง API ที่ใช้เพื่อ fetch ข้อมูลและการใช้ RESTful	43
ภาพที่ 3.3.3.6.1 ตัวอย่างการทำ Credentials สำหรับ Authentication	44
ภาพที่ 3.3.3.6.2 ตัวอย่างการกำหนด Session	44
ภาพที่ 3.3.3.6.3 ตัวอย่างการเรียกใช้ session เพื่อดึงข้อมูล user	45
ภาพที่ 4.1.1.1 แสดงหน้าสมัครบัญชีใหม่สำหรับเข้าสู่ระบบและตัวอย่าง Code	49
ภาพที่ 4.1.1.2 แสดงหน้าเข้าสู่ระบบ การอกรอบแบบและตัวอย่าง Code	50

ภาพที่ 4.1.2 แสดงหน้าสลับระหว่างเป็นผู้ซื้อกับเป็นผู้ขายและตัวอย่าง Code	51
ภาพที่ 4.1.3 แสดงหน้าฟอร์มรีวิวและหน้าสินค้าที่มีรีวิวและตัวอย่าง Code	52
ภาพที่ 4.1.4 แสดงหน้าดูประวัติการซื้อและตัวอย่าง Code	53
ภาพที่ 4.1.5 แสดงหน้าสินค้าที่สามารถเพิ่มสินค้าลงตะกร้าและตัวอย่าง Code	54
ภาพที่ 4.1.6 แสดงหน้าคิดตังและตัวอย่าง Code	55
ภาพที่ 4.1.7 แสดงหน้าหลักที่สามารถค้นหาหรือจัดเรียงสินค้าได้และตัวอย่าง Code	56
ภาพที่ 4.1.8 แสดงหน้าดูอเดอร์สินค้าของเกษตรกรและตัวอย่าง Code	57
ภาพที่ 4.1.9 แสดงหน้าร้านค้าที่สามารถเพิ่มสินค้าได้และตัวอย่าง Code	58
ภาพที่ 4.1.10 แสดงหน้าชุมชนและตัวอย่าง Code	59

## รายการสัญลักษณ์และคำย่อ

สัญลักษณ์/คำย่อ	คำเต็ม/คำจำกัดความ
4Ps	Product, Price, Place, Promotion
WTO	World Trade Organization
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
ORM	Object Relational Mapping
AWS	Amazon Web Services
SSR	Server-Side Rendering
SSG	Static Site Generation
SEO	Search Engine Optimization
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
UI	User Interface
UX	User Experience
UC	Use Case
IDE	integrated development environment
JWT	JSON Web Tokens

(13)

## บทที่ 1 บทนำ

### 1.1 ความเป็นมาและความสำคัญของโครงการ

เกษตรกรถือเป็นอาชีพที่มีจำนวนมากในประเทศไทย เนื่องจากพื้นที่ภูมิประเทศที่เหมาะสมต่อการเพาะปลูกและการประกอบอาชีพทางการเกษตร อย่างไรก็ตามในปัจจุบันอาชีพเกษตรกรกลับเป็นอาชีพที่ต้องเผชิญกับปัญหาอย่างต่อเนื่อง ไม่ว่าจะเป็นการถูกกดราคาจากพ่อค้าคนกลาง หรือการได้รับการสนับสนุนจากภาครัฐที่ไม่เพียงพอ ส่งผลให้เกษตรกรประสบความลำบากในการดำรงชีวิตและพัฒนาคุณภาพการผลิต ด้วยเหตุนี้ ผู้บริโภคจึงมีบทบาทสำคัญในการช่วยเหลือเกษตรกร ทั้งทางตรงและทางอ้อม หนึ่งในแนวทางที่สามารถแก้ไขปัญหาได้ คือ การเพิ่มช่องทางการค้าขายที่ช่วยให้เกษตรกรสามารถเข้าถึงผู้บริโภคได้โดยตรง ลดการพึ่งพาพ่อค้าคนกลาง และช่วยให้ผู้บริโภคสามารถเข้าถึงสินค้าจากเกษตรกรได้สะดวกมากยิ่งขึ้น ทีมของเรา มีความเชี่ยวชาญด้านการพัฒนาเทคโนโลยีและการเขียนโปรแกรม จึงมีแนวคิดที่จะนำความรู้นี้มาใช้ในการพัฒนาแพลตฟอร์มการซื้อขายสินค้าเกษตรโดยตรง เพื่อเป็นช่องทางที่เกษตรกรสามารถใช้งานได้ง่าย สะดวก และสอดคล้องกับวิถีชีวิตในยุคปัจจุบัน โดยแพลตฟอร์มนี้จะช่วยให้ผู้บริโภคสามารถศึกษาข้อมูลของเจ้าของฟาร์มและกระบวนการผลิตได้โดยตรง รวมถึงมีระบบตรวจสอบราคากลางแบบเรียลไทม์ ซึ่งช่วยให้ทั้งเกษตรกรและผู้บริโภคสามารถรับรู้ข้อมูลที่ทันสมัย โปร่งใส และแม่นยำ แพลตฟอร์มนี้เป็นการแก้ปัญหาที่ตอบโจทย์ยุคสมัยใหม่ โดยเฉพาะเกษตรกรรุ่นใหม่ที่มีการเข้าถึงและใช้งานเทคโนโลยีอย่างกว้างขวาง อันจะช่วยเสริมสร้างความยั่งยืนให้กับภาคการเกษตรของประเทศไทยในระยะยาว

### 1.2 วัตถุประสงค์

1. เพื่อช่วยเหลือเกษตรกรเรื่องการถูกพ่อค้าคนกลางกดราคา
2. เพื่อช่วยเพิ่มทางเลือกตลาดซื้อขายมากขึ้นให้เกษตรกรเพิ่มความมั่นคงให้เกษตรกร
3. เพื่อช่วยผู้บริโภคสามารถเลือกซื้อสินค้าเกษตรที่ตรงตามมาตรฐานตามความต้องการของผู้บริโภค
4. เพื่อเพิ่มปฏิสัมพันธ์ระหว่างตัวเกษตรตัวยกันและเกษตรกรกับผู้บริโภคมากยิ่งขึ้น
5. เพื่อเรียนรู้การสร้างระบบฐานข้อมูลที่มีประสิทธิภาพและปลอดภัย

6. เพื่อเรียนรู้การพัฒนาฟีเจอร์การจัดการระบบชำระเงินออนไลน์

### 1.3 ขอบเขตของโครงการ

1. ระบบต้องสามารถสร้างบัญชีผู้ใช้และสร้างประวัติโดยย่อผู้ขายและจัดการประวัติโดยย่อส่วนตัว
2. ระบบต้องสามารถแสดงรายการสินค้าของเกษตรกรที่ลงขายได้
3. ระบบต้องมีระบบการจัดการคำสั่งซื้อได้
4. ระบบต้องมีระบบวิจารณ์และการให้คะแนนสินค้าและส่วนของเกษตรกร
5. ระบบต้องค้นหาสินค้าและการจัดหมวดหมู่สินค้าได้
6. ระบบต้องมีช่องทางที่ให้ผู้ใช้ปฏิสัมพันธ์ระหว่างตัวเกษตรกรด้วยกันและเกษตรกรกับผู้บริโภค

### 1.4 ประโยชน์ของโครงการ

1. ได้ช่วยเหลือเกษตรกรเรื่องการถูกพ่อค้าคนกลางกดราคา
2. เกษตรกรสามารถขายสินค้าโดยตรงถึงผู้บริโภค ช่วยลดต้นทุน และเพิ่มรายได้
3. ได้เพิ่มทางเลือกตลาดซื้อขายมากขึ้นให้เกษตรกรเพิ่มความมั่นคงให้เกษตรกร
4. ได้ช่วยให้ผู้บริโภคสามารถเลือกซื้อสินค้าเกษตรที่ตรงตามความต้องการของผู้บริโภค
5. ได้เพิ่มปฏิสัมพันธ์ระหว่างตัวเกษตรกรด้วยกันและเกษตรกรกับผู้บริโภคมากยิ่งขึ้น
6. ผู้พัฒนาได้เรียนรู้กระบวนการพัฒนาเว็บทั้งฝั่ง Frontend และ Backend แบบครบวงจร

### 1.5 ข้อจำกัดของโครงการ

1. ข้อจำกัดด้านการเข้าถึงเทคโนโลยีของเกษตรกรที่ต้องใช้งานผ่านคอมพิวเตอร์เท่านั้น
2. ข้อจำกัดด้านความรู้และทักษะการใช้งานแพลตฟอร์มออนไลน์ของเกษตรกร
3. ข้อจำกัดด้านการแข่งขันกับตลาดแบบดั้งเดิม
4. ข้อจำกัดด้านความน่าเชื่อถือของชื่อฉลากราคาและตลาด
5. ข้อจำกัดด้านความหลากหลายของสินค้าในช่วงเริ่มต้น

## บทที่ 2

### วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

#### 2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 การตัดสินใจของผู้บริโภคต่อการซื้อสินค้าเกษตรออนไลน์

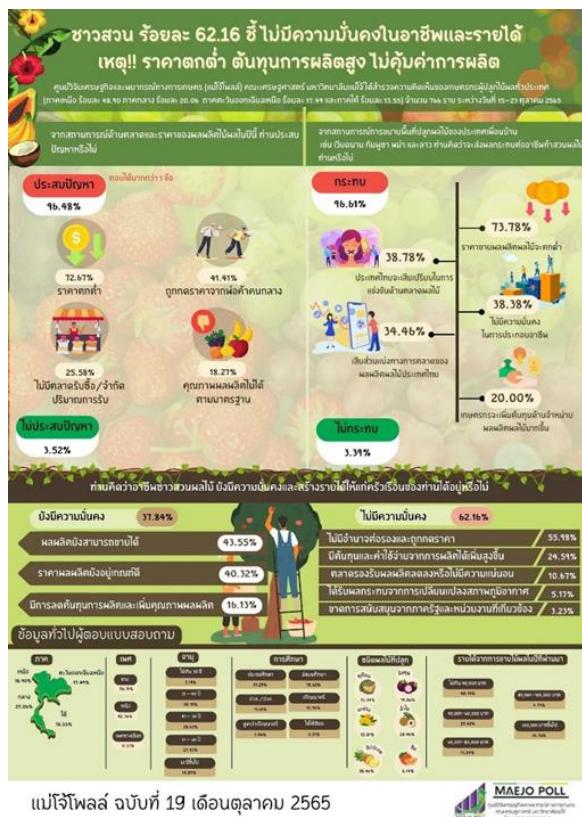
ผู้บริโภคร้อยละ 52.3 มีความเห็นว่า ปัญหาในการซื้อสินค้าเกษตรออนไลน์ คือ ไม่สามารถตรวจสอบสินค้าหรือสัมผัสสินค้าก่อนการซื้อได้ เนื่องจากผู้บริโภคไม่สามารถเห็นสินค้าจริง ก่อนตัดสินใจซื้อ ทำให้ผู้บริโภคบางรายไม่มั่นใจว่าจะได้รับสินค้าหลังจากชำระเงิน หรืออาจได้รับ สินค้าล่าช้า และไม่สามารถเปลี่ยนได้ จึงทำให้ผู้บริโภคไม่มั่นใจในการซื้อสินค้าเกษตรออนไลน์ ผู้บริโภคที่ซื้อสินค้าเกษตรออนไลน์ มีอายุ เฉลี่ย 32 ปี รายได้เฉลี่ย 25,788.56 บาทต่อเดือน ส่วน ใหญ่เลือกซื้อผลไม้แปรรูป โดยซื้อผ่าน Facebook ซึ่ง ซื้อนาน ๆ ครั้ง/ไม่ได้ซื้อทุกเดือน มีค่าใช้จ่ายในการซื้อ เฉลี่ย 673.35 บาทต่อครั้ง และชำระเงินผ่าน mobile banking ผู้บริโภคเห็นด้วยต่อการซื้อ สินค้าเกษตร ออนไลน์เนื่องจากมีความสะดวกและประหยัดเวลาในการเดินทาง ซึ่งผู้บริโภคค้นหา ข้อมูลสินค้าก่อนการซื้อ และตัดสินใจซื้อสินค้าเกษตรออนไลน์เนื่องจากมีการ จัดส่งสินค้าตรงตาม เวลา และไม่ต้องเสียเวลาเดินทาง ไปซื้อสินค้าเกษตรที่ร้าน(Agricultural Sci. J. 2021)

จากบทความนี้บพกความนี้มีความเกี่ยวข้องกับการพัฒนาแพลตฟอร์มซื้อขายสินค้าเกษตรออนไลน์ในหลายมิติ โดยเน้นไปที่ปัญหา ความต้องการ และพฤติกรรมของผู้บริโภค ซึ่งสามารถนำไปใช้ในการออกแบบและพัฒนาแพลตฟอร์ม เช่นปัญหาและข้อกังวลของผู้บริโภคไม่สามารถตรวจสอบหรือสัมผัสสินค้าก่อนการซื้อ ซึ่งทางที่ผู้บริโภคใช้ซื้อสินค้าเกษตร ประเภทสินค้า และพฤติกรรมการซื้อ เป็นต้นซึ่งข้อมูลจากบทความนี้สามารถนำไปปรับปรุงแพลตฟอร์มให้ตอบโจทย์ ทั้งในด้านเทคนิค การออกแบบ และการสร้างความเชื่อมั่นในการซื้อสินค้าเกษตรออนไลน์ ซึ่งจะช่วย เพิ่มประสิทธิภาพและความน่าสนใจของแพลตฟอร์มในระยะยาว

##### 2.1.2 ชาวสวนไม่มีความมั่นคงในอาชีพ

สำหรับด้านความมั่นคงทางอาชีพและรายได้ของชาวสวนผลไม้ พบร่วม ร้อยละ 62.16 คิดว่าไม่มีความมั่นคงทางอาชีพและรายได้ โดยให้เหตุผลว่าเนื่องจาก ไม่มีอำนาจต่อรองและถูกกด ราคาจากพ่อค้าคนกลาง ส่งผลให้ราคาผลผลิตตกต่ำ (ร้อยละ 55.98) รองลงมาคือ ค่าใช้จ่ายในการ ผลิตและต้นทุนจากการผลิตในปัจจุบันได้เพิ่มสูงขึ้น ส่งผลให้ไม่คุ้มค่าต่อการผลิต (ร้อยละ 24.59)

อันดับ 3 ตลาดรองรับผลผลิตลดลงหรือไม่มีความแน่นอนทั้งตลาดในประเทศและต่างประเทศ (ร้อยละ 10.67) ในขณะที่อีกร้อยละ 37.84 ยังคิดว่าอาชีพชาวสวนผลไม้ยังมีความมั่นคง เนื่องจากผลผลิตยังสามารถขายได้ทั้งตลาดภายในชุมชนและยังมีความต้องการของตลาดภายในประเทศ รวมถึงชาวสวนยังมีความพึงพอใจต่อราคากลางๆ(maejopoll,2022)



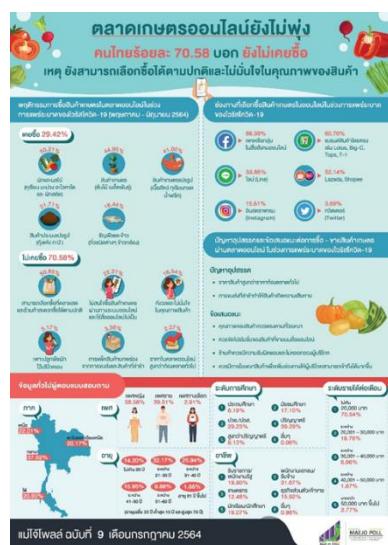
ภาพที่ 2.1.2 แสดงแผนภาพแสดงผลโพลอนากตชาวนานา民族ไม่สามารถในการ  
แข่งขันและความมั่นคงทางอาชีพ

ข้อมูลจากบทความนี้เน้นย้ำถึงความสำคัญของการสร้างแพลตฟอร์มที่สามารถแก้ปัญหารือว่างานจะต่อรองราคางานชิวส่วน การลดต้นทุน และการขยายตลาด ซึ่งจะช่วยให้ชิวส่วนมีรายได้ที่มั่นคงขึ้น และเสริมสร้างความยั่งยืนในอาชีพได้ในระยะยาว ซึ่งสามารถนำไปปรับปรุงฟังก์ชันและแนวทางการพัฒนาแพลตฟอร์ม เช่นการเพิ่มงานจต่อรองให้เกษตรกรสร้างตลาดกลางออนไลน์ที่เปิดกว้าง โปร่งใส แสดงราคาเฉลี่ยของตลาดแบบเรียลไทม์ ช่วยให้เกษตรกรตั้งราคาได้อย่างเหมาะสม ให้ข้อมูลเกี่ยวกับการลดต้นทุนการผลิตและแนวโน้มราคาวัตถุดิบ พัฒนาแพลตฟอร์มที่สามารถรองรับการซื้อขายในระดับนานาชาติ โดยมีระบบแปลงภาษาและช่องทางการชำระเงินหลากหลายสกุลเงินสร้างเครือข่ายพันธมิตรทั่วโลกและต่างประเทศ

### 2.1.3 ตลาดเกษตรออนไลน์ยังไม่มีแนวโน้มเติบโตสูงขึ้น

จากสภาพสินค้าที่มีลักษณะเฉพาะของสินค้าเกษตรทำให้ผลการสำรวจพบว่า มีผู้ใช้บริการซื้อสินค้าเกษตรผ่านระบบออนไลน์ไม่มากนักเนื่องจากผู้บริโภคกับพบเจอบัญหาในคุณภาพของสินค้าที่มักเกิดความเสียหายจากการขนส่งที่ล่าช้าหรือคุณภาพของสินค้าที่ไม่ตรงตามคุณลักษณะที่ร้านค้าหรือผู้ขายได้ระบุไว้ จึงควรมีหน่วยงานที่เกี่ยวข้องดูแลตรวจสอบคุณภาพของสินค้า และ มีการรับรองในคุณภาพของสินค้าจากผู้ขายไปจนกว่าจะถึงมือผู้บริโภคเพื่อสร้างความมั่นใจในคุณภาพของสินค้าให้แก่ผู้บริโภคต่อไปในอนาคต(maejopoll,2021)

บทความนี้ชี้ให้เห็นถึงปัญหาที่สำคัญในการซื้อสินค้าเกษตรออนไลน์ ซึ่งเกี่ยวข้องกับ การรักษาคุณภาพสินค้าและการขนส่ง แพลตฟอร์มควรเน้นการตรวจสอบคุณภาพสินค้า การขนส่งที่ มีประสิทธิภาพ และการสร้างความโปร่งใส เพื่อสร้างความมั่นใจให้กับผู้บริโภค ซึ่งจะช่วยเพิ่มความ น่าเชื่อถือ และส่งเสริมให้มีการใช้งานแพลตฟอร์มมากขึ้นในอนาคต ซึ่งสามารถนำข้อมูลนี้มาปรับปรุง พัฒนาการทำงานของแพลตฟอร์มได้ เช่น แสดงตราสักรองคุณภาพสินค้าบนหน้าสินค้า สร้างระบบ ติดตามการขนส่งแบบเรียลไทม์ เสนอการขนส่งแบบควบคุมอุณหภูมิ (Cold Chain Logistics) สำหรับสินค้าที่ต้องการการดูแลเป็นพิเศษ



ภาพที่ 2.2.3 แสดงแผนภาพแสดงผลโอลโคลิโวิด-19 กับตลาดสินค้าเกษตรยุคดิจิทัล

#### 2.1.4 ปัจจัยที่มีผลต่อการตัดสินใจเลือกซื้อผักและผลไม้สดของผู้บริโภคในระบบ สายโซ่ความเย็น(Cold Chain)

ปัจจัยที่มีผลต่อการตัดสินใจเลือกซื้อผักและผลไม้สดของผู้บริโภคในระบบสายโซ่ความเย็น (Cold Chain) จากร้านค้าปลีกสมัยใหม่ เป็นการวิจัยเชิงปริมาณ มีวัตถุประสงค์เพื่อศึกษา พฤติกรรมการเลือกซื้อ ปัญหาและอุปสรรคในการเลือกซื้อ รวมถึงปัจจัยด้านส่วนประสมทาง การตลาด (4Ps) ที่มีผลต่อการตัดสินใจเลือกซื้อผักและผลไม้สดในระบบสายโซ่ความเย็น (Cold Chain) ของผู้บริโภคในเขตกรุงเทพมหานคร และปริมณฑลจำนวน 400 คน (มาริสา, 2019)

บทความนี้ให้ข้อมูลที่สำคัญเกี่ยวกับพฤติกรรมการซื้อสินค้าเกษตรสด และ ปัจจัยที่มีผลต่อการตัดสินใจซื้อ ซึ่งสามารถนำมาใช้ในการพัฒนาแพลตฟอร์ม โดยเน้นที่การมีระบบ Cold Chain ที่มีประสิทธิภาพ และการปรับใช้กลยุทธ์การตลาด (4Ps) เพื่อเพิ่มความมั่นใจและความ พึงพอใจให้กับผู้บริโภคในตลาดเมืองใหญ่ เช่น กรุงเทพฯ และปริมณฑล

### 2.1.5 มาตรฐานสินค้าเกษตรสำคัญอย่างไร และมีอะไรบ้าง

อาหารเป็นหนึ่งในปัจจัยสี่ที่ช่วยหล่อเลี้ยงชีวิตมนุษย์ ในอดีตมนุษย์ได้อาหารมาจาก การเก็บของป่า ล่า สัตว์ และเกษตรกรรม ปัจจุบันพัฒนาจากอาหารส่วนใหญ่ที่เรารับริโภค ผลิตจากอุตสาหกรรมอาหาร จากการขยายตัวทางเศรษฐกิจ มีการนำเข้าและส่งออกอาหารระหว่างประเทศมากมาย จึงเป็นปัจจัย ผลักดันให้เกิดการแข่งขันด้านคุณภาพของสินค้าและการบริการที่สูงขึ้น ประกอบกับการปฏิบัติตาม กรอบกติกาการค้าเสรี ของ World Trade Organization (WTO) จำเป็นต้องมีการพัฒนาระบบการ ผลิต และการตรวจสอบเพื่อออกใบปรับปรุงที่ได้คุณภาพตามมาตรฐานสากล เพื่อสร้างความน่าเชื่อถือ และเพื่อตอบสนองความต้องการของผู้บริโภค(mitrpholmodernfarm,2022)

บทความนี้ชี้ให้เห็นถึงความสำคัญของการรักษามาตรฐานคุณภาพในสินค้าเกษตร และการ ตอบสนองต่อการแข่งขันในตลาดโลก ซึ่งแพลตฟอร์มชี้อิยาลสินค้าเกษตรสามารถนำไปปรับใช้ได้โดย การพัฒนาระบบตรวจสอบคุณภาพ การแสดงข้อมูลรับรอง และการเพิ่มช่องทางการเข้าถึงตลาด สากล ช่วยสร้างความน่าเชื่อถือและตอบสนองความต้องการของผู้บริโภคได้อย่างมีประสิทธิภาพ



ภาพที่ 2.1.5 แสดงสัญลักษณ์ของ WTO

## 2.2 งานวิจัยที่เกี่ยวข้อง

### 2.2.1 DGT Farm

โดยสำนักงานมาตรฐานเกษตรและอาหารแห่งชาติ หรือ มกอช. กับเว็บไซต์ [www.DGTFarm.com](http://www.DGTFarm.com) เป็นตลาดกลางให้เกษตรกร และผู้บริโภคได้ซื้อสินค้า ซึ่งปัจจุบันมีเกษตรและกลุ่มวิสาหกิจชุมชนกว่า 1,006 ราย มีสินค้าให้เลือกกว่า 553 รายการ มีทั้ง ข้าว ผักผลไม้ สินค้าเกษตรแปรรูป (rakbankerd,2020)

โดยตัวเว็บไซต์ DGTFarm นั้นมีฟังก์ชั่นการใช้งานหลายอย่างคล้ายกับที่เราได้วางแผนที่จะพัฒนาระบบเอาไว้ไม่ว่าจะเป็นฟังก์ชั่นช่วยการันตีคุณภาพของสินค้า การเกษตรตามคุณภาพของสินค้า เกษตรนั้นหรือฟังก์ชั่นแบ่งแยกประเภทสินค้าช่วยค้นหาสินค้าตามความต้องการของลูกค้า แต่ก็ยังมีอีกหลายฟังก์ชั่นที่สามารถเพิ่มและปรับปรุงพัฒนาได้ เช่นระบบของเราจะมีฟังก์ชั่นที่จะทำให้เกษตรกรได้รู้ราคาเป็นไปของตลาดได้ การปรับปรุงฟังก์ชั่นการค้นหาให้สามารถเฉพาะเจาะจงได้ตามความต้องการ ฟังก์ชั่นคอมมูนิตี้ของเกษตรกรเป็นต้น



ภาพที่ 2.2.1 แสดงหน้าตาของเว็บไซต์ DGT Farm

## 2.2.2 JatujakMall



ภาพที่ 2.2.2 แสดงหน้าตาของเว็บไซต์ JatujakMall

แพลตฟอร์มนี้เพื่อมุ่งเน้นยกระดับมาตรฐานด้านคุณภาพในกรุงเทพมหานคร เป็นตลาดที่ได้รับความนิยมจากคนไทยและชาวต่างชาติ มาอยู่ในรูปแบบออนไลน์ เพื่อกระจายสินค้า ทั่วไทยและส่งออกไปทั่วโลก ซึ่งนำเสนอวิธีการที่ช่วยอำนวยความสะดวกให้กับ Smart Farmer Online ทุกคนด้วยบริการจัดทำข้อมูล ถ่ายภาพและวิดีโอ เพื่อโปรโมตสินค้า(rakbankerd,2020)

โดยแพลตฟอร์ม JatujakMall ก็เป็นอีกช่องทางที่เป็นเหมือนตลาดที่ให้ชาวเกษตรสามารถขายสินค้าเกษตร เป็นเหมือนกับตลาดทางเลือกอีกแพลตฟอร์มแต่ก็ยังมีหลายพื้นที่ที่ไม่เฉพาะเจาะจงหรือไม่มีรายเท่ากับระบบที่เราจะพัฒนาเช่นไม่มีการค้นสินค้าตามมาตรฐาน การเกษตร ไม่มีระบบต้องมีไฟเจอร์การวิเคราะห์ข้อมูล เป็นต้น

## 2.3 เทคโนโลยีที่เกี่ยวข้องกับโครงการ

โครงการนี้พัฒนาขึ้นโดยใช้เทคโนโลยีที่หลากหลาย เพื่อให้ระบบมีประสิทธิภาพและสามารถตอบสนองความต้องการของผู้ใช้งานได้อย่างสมบูรณ์แบบ โดยรายละเอียดของเทคโนโลยีที่เกี่ยวข้องมีดังนี้:

### 2.3.1 HTML (HyperText Markup Language)

HTML เป็นภาษาマーกอปที่ใช้ในการสร้างโครงสร้างพื้นฐานของหน้าเว็บ โดยทำหน้าที่กำหนดองค์ประกอบต่าง ๆ เช่น ข้อความ รูปภาพ ลิงก์ และตาราง ซึ่งเป็นส่วนสำคัญในการออกแบบและพัฒนาอินเทอร์เฟซของผู้ใช้ (User Interface) (Brennan Whitfield, 2025)

โครงการนี้ใช้เทคโนโลยีในการพัฒนาเว็บแอปพลิเคชัน โดยใช้ Next.js ซึ่งเป็นเฟรมเวิร์กของ React.js ที่รองรับการเขียนโค้ดด้วยภาษา JavaScript และ JSX ซึ่งจะถูกคอมไพล์เป็น HTML และ JavaScript เพื่อแสดงผลบนเบราว์เซอร์ ดังนั้น HTML จึงเป็นส่วนหนึ่งของระบบที่ใช้ในการแสดงผลหน้าเว็บไซต์



ภาพที่ 2.3.1 แสดงสัญลักษณ์ของ HTML

### 2.3.2 CSS (Cascading Style Sheets)

CSS นั้นจะทำหน้าที่บอกถึงรูปลักษณ์การแสดงผลของ HTML Element โดยกำหนด Attribute (คุณสมบัติ) ต่างๆ เพื่อให้แสดงผลตามที่เราต้องการ เป็นภาษาโปรแกรมมิ่งที่สำคัญในการตกแต่งหน้าตาของเว็บไซต์โดยทีเดียว (Sirisak Sueakkam,2024)

ในโครงงานนี้ CSS ถูกใช้ร่วมกับ TailwindCSS ใน การตกแต่งหน้าเว็บไซต์ e-commerce ให้มีความสวยงาม ทันสมัย และมีการจัดวางที่เหมาะสมกับอุปกรณ์ที่หลากหลาย โดยมีการกำหนดสีของปุ่ม รูปแบบข้อความ และการจัด layout ของแต่ละ section ผ่าน class utility ต่าง ๆ



ภาพที่ 2.3.2 แสดงสัญลักษณ์ของ CSS

### 2.3.3 Next.js Framework

Next.js เป็นเฟรมเวิร์กที่พัฒนาบนพื้นฐานของ React ช่วยในการพัฒนาเว็บแอปพลิเคชันที่มีประสิทธิภาพสูง รองรับการเรนเดอร์หน้าเว็บแบบเซิร์ฟเวอร์ไซต์ (Server-Side Rendering) และการเรนเดอร์แบบ Static(Asian Digital Hub,2024)

ในโครงงานนี้ Next.js เป็นโครงหลักของระบบ ใช้สำหรับสร้างหน้าเว็บต่าง ๆ เช่น หน้า Homepage, หน้า Login/Register, หน้าแสดงสินค้า, หน้า Profile, หน้า Community และ API Route สำหรับเชื่อมต่อกับฐานข้อมูล ผ่านการจัดโครงสร้างแบบ App Router ที่ช่วยให้แยกไฟล์แต่ละหน้าได้อย่างชัดเจน



ภาพที่ 2.3.3 แสดงสัญลักษณ์ของ Next.js

#### 2.3.4 PostgreSQL

เป็นระบบฐานข้อมูลเชิงสัมพันธ์แบบโอลีเพ่นชอร์สอันทรงพลังที่ใช้และขยายภาษา SQL ร่วมกับฟีเจอร์มากมายที่จัดเก็บและปรับขนาดเวิร์กโหลดข้อมูลที่ซับซ้อนที่สุดอย่างปลอดภัย (PostgreSQL, n.d.)

ในโครงงานนี้ PostgreSQL ทำหน้าที่เป็นฐานข้อมูลหลักของระบบ ใช้เก็บข้อมูลผู้ใช้งาน สินค้า คำสั่งซื้อ ที่อยู่ การรีวิว รวมถึงโพสต์และคอมเม้นต์ใน community โดยมีการออกแบบ schema เพื่อรับความสัมพันธ์ระหว่างข้อมูลแต่ละประเภท



ภาพที่ 2.3.4 แสดงสัญลักษณ์ของ PostgreSQL

### 2.3.5 Prisma ORM (Object Relational Mapping)

คือเครื่องมือจัดการฐานข้อมูล open source สำหรับ application TypeScript และ JavaScript โดย Prisma ORM จะมีชุดเครื่องมือในการสร้างและจัดการโครงสร้างของฐานข้อมูล สั่งคำสั่งค้นหา (queries) รวมถึงสามารถโยกย้ายฐานข้อมูล (migrations) ได้อย่างมีประสิทธิภาพและใช้งานง่าย(mikelopster,2024)

ในโครงงานนี้ Prisma ถูกใช้ในการเขียนคำสั่งเพิ่ม ลบ แก้ไข และตีงข้อมูล เช่น การดึงรายการสินค้า, เพิ่มคำสั่งซื้อ, แสดงประวัติการสั่งซื้อ, หรืออัปเดตข้อมูลผู้ใช้ โดยไม่ต้องเขียน SQL ตรง ๆ ซึ่งลดโอกาสเกิด error และทำให้โค้ดอ่านง่ายขึ้น



ภาพที่ 2.3.5 แสดงสัญลักษณ์ของ Prisma ORM

### 2.3.6 GitHub

GitHub เป็นแพลตฟอร์มสำหรับการจัดเก็บและจัดการซอฟต์แวร์ด้านของการ โดยรองรับการทำงานร่วมกันของนักพัฒนา ช่วยในการควบคุมเวอร์ชันของโค้ดและติดตามการเปลี่ยนแปลง (Ben Lutkevich,2024)

ในโครงงานนี้ GitHub ถูกใช้เป็นที่จัดเก็บโค้ดทั้งหมดของโปรเจกต์



ภาพที่ 2.3.6 แสดงสัญลักษณ์ของ Github

### 2.3.7 TailwindCSS



ภาพที่ 2.3.7 แสดงสัญลักษณ์ของ TailwindCSS

Tailwind CSS คือ CSS Framework ตัวหนึ่งที่ใช้งานแบบ Utility-first หมายถึงแต่ละ class จะมีคุณสมบัติอยู่ ๆ ของ CSS เก็บในตัวของมันอย่างละเอียด เช่น Background-color, Border, font-size ฯลฯ ซึ่งเหมาะกับคนที่ต้องการปรับแต่งหน้าเว็บอย่างอิสระ เพราะแต่ละ class นั้นเก็บคำสั่ง CSS ไว้แค่อย่างละเอียด ไม่ใช่ Class component สำเร็จรูปที่รวมหลาย ๆ ชุดคำสั่ง เราสามารถเข้าไป (Kittikorn Voraprateep, 2023)

### 2.3.8 supabase

Supabase คือ open source platform ที่ทำการเตรียมเครื่องมือให้นักพัฒนาสามารถเขียน application ได้อย่างรวดเร็วขึ้น ซึ่งหลาย ๆ คนจะมองว่า Supabase มีลักษณะใกล้เคียงกับ Firebase (mikelopste, 2024)

ถูกนำมาใช้ในส่วนของ การอัปโหลดและจัดเก็บไฟล์รูปภาพสินค้า หรือ รูปโปรไฟล์ ผู้ใช้งาน แทนการจัดเก็บไฟล์ไว้ในเครื่องหรือระบบ local โดยสามารถเข้าถึงและจัดการไฟล์ผ่าน API ได้สะดวก และยังช่วยลดภาระบน server หลักของระบบ



ภาพที่ 2.3.8 แสดงสัญลักษณ์ของ supabase

## บทที่ 3

### วิธีการวิจัย

#### 3.1 ภาพรวมของโครงการ

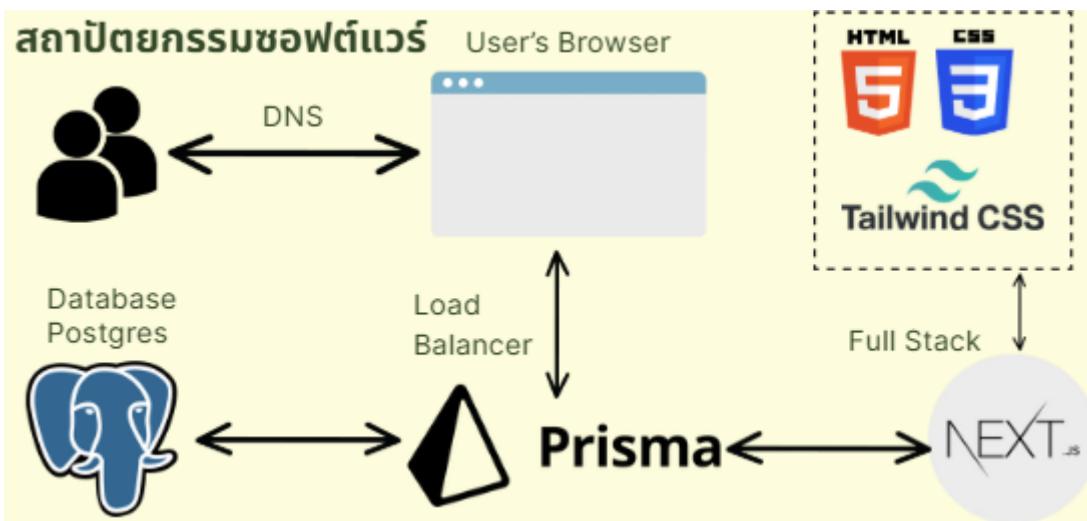
โครงการนี้มีวัตถุประสงค์เพื่อพัฒนา แพลตฟอร์มซื้อขายสินค้าเกษตรออนไลน์ ที่มุ่งเน้นแก่ผู้ผลิตและตอบสนองความต้องการของทั้งผู้บริโภคและเกษตรกร โดยแพลตฟอร์มถูกออกแบบให้รองรับการซื้อขายที่มีความหลากหลาย ปลอดภัย และโปร่งใส ทั้งยังช่วยลดข้อกังวลเรื่องคุณภาพของสินค้า การขนส่ง และราคาที่ไม่แน่นอน ซึ่งเป็นปัจจัยสำคัญที่ส่งผลต่อพฤติกรรมของผู้บริโภคและรายได้ของเกษตรกร

ระบบนี้ใช้สถาปัตยกรรมแบบ Modular Design ที่แบ่งการทำงานออกเป็นเลเยอร์ต่างๆ เพื่อเพิ่มประสิทธิภาพ ความยืดหยุ่น และความสามารถในการปรับขยาย โดยโครงการนี้ประกอบด้วย 3 เลเยอร์หลัก ได้แก่ Frontend Layer, Backend Layer, Database Layer ซึ่งแต่ละเลเยอร์ทำงานร่วมกันเพื่อสนับสนุนการใช้งานของผู้ใช้ และการจัดการธุกรรมในระบบอย่างครบถ้วน

การออกแบบสถาปัตยกรรมของระบบถูกพัฒนาขึ้นโดยคำนึงถึงความสะดวกในการใช้งาน ความปลอดภัยของข้อมูล และการประมวลผลที่รวดเร็ว พิจารณาด้วยตัวในอนาคต เช่น การเพิ่มจำนวนผู้ใช้ การเชื่อมต่อกับตลาดสากล และการรองรับเทคโนโลยีใหม่

### 3.1.1 สถาปัตยกรรมของระบบ

โครงสร้างของสถาปัตยกรรมระบบแบ่งออกเป็น 4 เลเยอร์หลัก ซึ่งอธิบายการทำงานในแต่ละส่วนดังนี้:



ภาพที่ 3.1.1 แสดงสถาปัตยกรรมของระบบ

#### 3.1.1.1 Frontend Layer

Frontend Layer คือส่วนของระบบที่ใช้แสดงผลและโต้ตอบกับผู้ใช้งาน ซึ่งถือเป็นส่วนสำคัญในการสร้างประสบการณ์ที่ดีและดึงดูดใจผู้ใช้งาน (User Experience)

- Next.js

Next.js ถูกเลือกเป็นเฟรมเวิร์กหลักในการพัฒนา Frontend เนื่องจากรองรับ Server-Side Rendering (SSR) และ Static Site Generation (SSG) ซึ่งช่วยเพิ่มความเร็วในการโหลดหน้าเว็บ ทำให้หน้าเว็บสามารถแสดงผลข้อมูลได้อย่างรวดเร็ว และส่งผลดีต่อ SEO ตัวอย่างเช่น เมื่อผู้ใช้งานค้นหาสินค้าเกี่ยวกับเครื่องมือคันหา หน้าเว็บจะแสดงข้อมูลเช่น ชื่อสินค้า ราคาสินค้า และรีวิวผู้ใช้งานได้ทันทีโดยไม่ต้องรอโหลดเพิ่มเติม

- Tailwind CSS

Tailwind CSS ถูกนำมาใช้ในการออกแบบและจัดวางหน้าตาของระบบ โดยเป็นเฟรมเวิร์กที่ช่วยสร้าง UI ที่ดูสะอาด ทันสมัย และตอบสนองต่ออุปกรณ์หลากหลาย (Responsive Design) ได้อย่างง่ายดาย ตัวอย่างเช่น การออกแบบหน้าสินค้าที่แสดงข้อมูลสำคัญอย่างชัดเจน ทั้งบนหน้าจอ มือถือ และเดสก์ท็อป

### 3.1.1.2 Backend Layer

Backend Layer ของโครงงานนี้เป็นส่วนที่รับผิดชอบในการประมวลผลข้อมูล จัดการฐานข้อมูล และติดต่อกับ Frontend ผ่าน API โดยใช้เทคโนโลยีและเครื่องมือต่าง ๆ ดังนี้:

-ภาษาและเฟรมเวิร์กที่ใช้:

Backend พัฒนาด้วยภาษา TypeScript โดยใช้เฟรมเวิร์ก Next.js ซึ่งรองรับการสร้าง API Route ภายใต้ตัว ทำให้สามารถพัฒนาเว็บไซต์และ API ภายใต้โครงสร้างเดียวกันได้อย่างมีประสิทธิภาพ

-การเชื่อมต่อกับ Frontend:

Frontend จะเรียกใช้ข้อมูลจาก Backend ผ่าน API ด้วย HTTP Request โดยข้อมูลที่ได้จะอยู่ในรูปแบบ JSON ซึ่งสามารถนำไปแสดงผลในเว็บไซต์ได้ทันที

-API Endpoint: Backend มีการสร้าง API Route สำหรับฟีเจอร์ต่าง ๆ เช่น:

/api/products: สำหรับดึงและเพิ่มข้อมูลสินค้า

/api/orders: สำหรับสร้างรายการสั่งซื้อและตรวจสอบสถานะ

/api/auth: สำหรับการยืนยันตัวตนของผู้ใช้งาน (Authentication)

### 3.1.1.3 Database Layer

ใช้ฐานข้อมูล PostgreSQL สำหรับจัดเก็บข้อมูลทั้งหมด เช่น ข้อมูลผู้ใช้งาน ร้านค้า สินค้า ออเดอร์ และรีวิว โดยใช้ Prisma เป็นเครื่องมือ ORM (Object-Relational Mapping) เพื่อช่วยในการติดต่อกับฐานข้อมูลให้สะดวกและปลอดภัยมากยิ่งขึ้น การออกแบบ Schema สำหรับตารางหลักต่าง ๆ เช่น:

User: เก็บข้อมูลผู้ใช้งาน

Store: เก็บข้อมูลร้านค้าของผู้ขาย

Product: ข้อมูลสินค้าเกษตรที่จำหน่าย

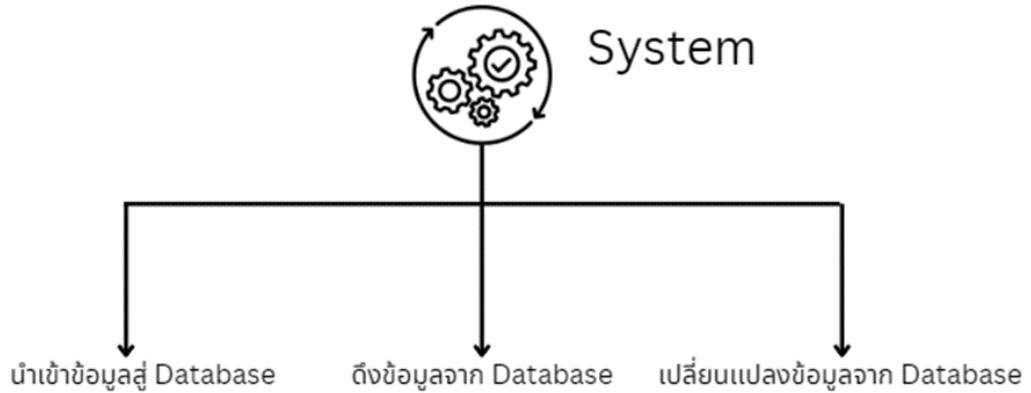
Order และ OrderItem: ใช้เก็บข้อมูลการสั่งซื้อ

Review: ความคิดเห็นและคะแนนจากลูกค้า

Address: ที่อยู่สำหรับจัดส่งสินค้า

### 3.2 การวิเคราะห์ขอบเขตและความต้องการของระบบ

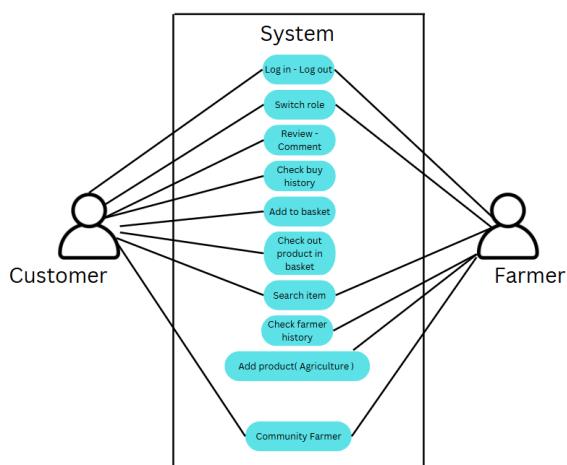
#### 3.2.1 วิเคราะห์ความต้องการของระบบในด้านของการทำงานด้วยตนเอง



ภาพที่ 3.2.1 แสดงความต้องการของระบบในด้านของการทำงานด้วยตนเอง

จากการวิเคราะห์ความต้องการของระบบ ระบบต้องสามารถนำข้อมูลใหม่ๆเข้าสู่ Database ได้ ระบบต้องสามารถนำข้อมูลจาก Database มาแสดงและนำมาใช้งานได้ ระบบมีความสามารถที่จะเปลี่ยนแปลงข้อมูลที่ได้รับมาจาก User มาอัพเดตข้อมูลใน Database

#### 3.2.2 วิเคราะห์ความต้องการของระบบในด้านของผู้ใช้งาน



ภาพที่ 3.2.2 แสดงความต้องการของระบบในด้านของผู้ใช้งาน

จากการวิเคราะห์ความต้องการของระบบ ผู้ใช้งานของระบบมี Actor คือ Customer และ Farmer ที่ Customer สามารถที่จะ Log in – Log Out มีความสามารถในการ Switch role ไปมา สามารถที่จะ Review สินค้าได้ และมีแค' Customer เท่านั้นที่สามารถ Comment สินค้าได้ สามารถจะเช็คประวัติการซื้อขายได้ สามารถนำสินค้าเข้าสู่ตั๋กร้าเพื่อซื้อขาย สามารถค้นหาสินค้าได้ ส่วน Farmer สามารถที่จะ Log in – Log Out มีความสามารถในการ Switch role ไปมา สามารถที่จะดู Review สินค้าได้ และ Farmer ยังสามารถตรวจสอบประวัติการขายของ Farmer เองได้ สามารถลงขายสินค้าลงเว็บไซต์ได้

Actor	คำอธิบาย
Customer	บุคคลที่มีความต้องการซื้อสินค้าเกษตรเพื่อการบริโภคส่วนตัวหรือเพื่อการพาณิชย์ สามารถเข้าถึงการค้นหาสินค้า การสั่งซื้อสินค้า การติดตามสถานะการจัดส่ง การให้คะแนนและรีวิว
Farmer	ผู้ผลิตสินค้าเกษตร เช่น ผัก ผลไม้ พืช หรือสินค้าแปรรูปที่มีความพร้อมในการจำหน่ายสินค้าผ่านระบบออนไลน์ สามารถเข้าถึงสร้างໂປຣไฟล์ผู้ขาย การเพิ่มและจัดการสินค้า ติดตามยอดขายและรายได้จากระบบ

### ตารางที่ 3.2.2 กรณีใช้งาน (Use Case) ต่าง ๆ

ลำดับ	ชื่อ Use Case	Actor	คำอธิบาย
UC01	Log in - Log out	-Customer -Farmer	สามารถเข้าสู่ระบบหรือออกจากระบบได้โดยใช้บัญชีที่ลงทะเบียนไว้ ซึ่งประกอบด้วยชื่อผู้ใช้และรหัสผ่าน หรือใช้ Social Media Login เพื่อเพิ่มความสะดวก
UC02	Switch role	-Customer -Farmer	ผู้ใช้ที่มีหลายบทบาท เช่น เป็นทั้ง Farmer และ Customer สามารถสลับบทบาทในระบบได้ โดยไม่ต้องสร้างบัญชีใหม่
UC03	Review - Comment	-Customer	ผู้ใช้สามารถให้คะแนนและเขียนรีวิวสินค้า รวมถึงแสดงความคิดเห็นเกี่ยวกับคุณภาพสินค้า การบริการ หรือการจัดส่ง
UC04	Check buy history	-Customer	สามารถตรวจสอบประวัติการซื้อสินค้า และติดตามสินค้าได้
UC05	Add to basket	-Customer	สามารถเพิ่มสินค้าที่ต้องการซื้อในตะกร้าสินค้า เพื่อร่วบรวมรายการสินค้าก่อนทำการชำระเงิน
UC06	Check out product in basket	-Customer	สามารถดำเนินการทำเงินสำหรับสินค้าที่อยู่ในตะกร้าสินค้า โดยเลือกวิธีชำระเงิน และวิธีจัดส่งที่ต้องการ
UC07	Search item	-Customer -Farmer	สามารถค้นหาสินค้าหรือรายการที่ต้องการในระบบ โดยใช้คำค้นหรือพิลเตอร์ เช่น ชื่อสินค้า ประเภท หรือราคา หรือมาตรฐานของสินค้า

UC08	Check farmer history	-Farmer	สามารถตรวจสอบประวัติการขายและแสดงสถิติต่างๆ
UC09	Add product( Agriculture )	-Farmer	สามารถเพิ่มสินค้าใหม่เข้าสู่ระบบ โดยระบุรายละเอียด เช่น ชื่อสินค้า ราคา ปริมาณ และรูปภาพ
UC10	Community Farmer	-Customer -Farmer	สามารถมีส่วนร่วมในชุมชนออนไลน์ เช่น การแลกเปลี่ยนความรู้ ทางการเกษตร การประกาศข่าวสาร หรือการหารือเรื่อง การพัฒนาผลิตภัณฑ์

ตารางที่ 3.2.2.1 กรณีใช้งาน (Use Case) Log in - Log out

Use Case Name	UC01:Log in - Log out
Primary Actor	Customer, Farmer
Goal	ผู้ใช้สามารถเข้าสู่ระบบและออกจากระบบได้
Precondition	ผู้ใช้ต้องมีบัญชีผู้ใช้งานที่ลงทะเบียนในระบบแล้ว
Main Flow	<ol style="list-style-type: none"> <li>1. ผู้ใช้กรอกชื่อผู้ใช้และรหัสผ่าน</li> <li>2. ระบบตรวจสอบข้อมูล</li> <li>3. ผู้ใช้เข้าสู่ระบบสำเร็จ</li> <li>4. ผู้ใช้ออกจากระบบสำเร็จ</li> </ol>
Alternate Flow	1a. หากข้อมูลผิดพลาด ระบบแจ้งเตือนให้ผู้ใช้ลองใหม่
Postcondition	ผู้ใช้สามารถเข้าถึงฟังก์ชันในระบบได้เมื่อเข้าสู่ระบบสำเร็จ
Exception	- หากลืมรหัสผ่าน ผู้ใช้สามารถกู้คืนได้

ตารางที่ 3.2.2.2 กรณีใช้งาน (Use Case) Switch role

Use Case Name	UC02: Switch role
Primary Actor	Customer, Farmer
Goal	ผู้ใช้สามารถสลับบทบาทระหว่าง Customer และ Farmer
Precondition	ผู้ใช้ต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> <li>1. ผู้ใช้กดปุ่ม Switch Role</li> <li>2. ระบบเปลี่ยนบทบาทการแสดงผลและฟังก์ชัน</li> </ol>
Postcondition	บทบาทของผู้ใช้เปลี่ยนไปตามที่เลือก

### ตารางที่ 3.2.2.3 กรณีใช้งาน (Use Case) Review - Comment

Use Case Name	UC03: Review - Comment
Primary Actor	Customer
Goal	ลูกค้าสามารถให้คะแนนและเขียนรีวิวสินค้า
Precondition	ลูกค้าใช้ต้องซื้อสินค้าก่อน
Main Flow	<ol style="list-style-type: none"> <li>1. ลูกค้าเลือกสินค้าที่ซื้อ</li> <li>2. กดปุ่มรีวิว</li> <li>3. ระบุคะแนนและเขียนความเห็น</li> <li>4. กดแสดงความคิดเห็น</li> </ol>
Postcondition	รีวิวถูกแสดงในหน้าสินค้า

### ตารางที่ 3.2.2.4 กรณีใช้งาน (Use Case) Check buy history

Use Case Name	UC04: Check buy history
Primary Actor	Customer
Goal	สามารถดูประวัติการซื้อสินค้าได้
Precondition	ลูกค้าต้องเข้าสู่ระบบ
Main Flow	<ol style="list-style-type: none"> <li>1. ลูกค้าเลือก "ประวัติการซื้อ"</li> <li>2. ระบบแสดงรายการสินค้าที่ซื้อสำเร็จ</li> <li>3. สามารถตรวจสอบสถานะของสินค้าที่สั่งซื้อ</li> </ol>
Postcondition	ลูกค้าสามารถดูรายการสินค้าที่ซื้อสำเร็จและตรวจสอบสถานะของสินค้า

### ตารางที่ 3.2.2.5 กรณีใช้งาน (Use Case) Add to basket

Use Case Name	UC05: Add to basket
Primary Actor	Customer
Goal	ลูกค้าเพิ่มสินค้าที่ต้องการซื้อไปยังตะกร้า
Precondition	ลูกค้าต้องเข้าสู่ระบบ
Main Flow	<ul style="list-style-type: none"> <li>1. ลูกค้าเลือกสินค้าที่ต้องการ</li> <li>2. กดปุ่ม "เพิ่มในตะกร้า"</li> </ul>
Postcondition	สินค้าปรากฏในตะกร้าสินค้า

### ตารางที่ 3.2.2.6 กรณีใช้งาน (Use Case) Check out product in basket

Use Case Name	UC06: Check out product in basket
Primary Actor	Customer
Goal	ลูกค้าสามารถชำระเงินและยืนยันการซื้อสินค้า
Precondition	ลูกค้าต้องมีสินค้าที่เพิ่มในตะกร้าสินค้า
Main Flow	<ul style="list-style-type: none"> <li>1. ลูกค้าเลือก "ชำระเงิน"</li> <li>2. กรอกข้อมูลการจัดส่งและวิธีชำระเงิน</li> <li>3. กดยืนยัน</li> </ul>
Postcondition	การสั่งซื้อสำเร็จและอยู่ในสถานะรอการจัดส่ง

### ตารางที่ 3.2.2.7 กรณีใช้งาน (Use Case) Search item

Use Case Name	UC07: Search item
Primary Actor	Customer, Farmer
Goal	ผู้ใช้สามารถค้นหาสินค้าในระบบได้
Precondition	ระบบต้องมีสินค้าพร้อมค้นหา
Main Flow	<ol style="list-style-type: none"> <li>1. ผู้ใช้กรอกคำค้นในช่องค้นหา</li> <li>2. ระบบแสดงผลลัพธ์ที่เกี่ยวข้อง</li> </ol>
Postcondition	ผู้ใช้พบรายการสินค้าที่ต้องการ

### ตารางที่ 3.2.2.8 กรณีใช้งาน (Use Case) Check farmer history

Use Case Name	UC08: Check farmer history
Primary Actor	Farmer
Goal	เกษตรกรสามารถตรวจสอบประวัติการขายของเกษตรกรได้
Precondition	เกษตรกรต้องมีรายการสินค้าที่ขายแล้ว
Main Flow	<ol style="list-style-type: none"> <li>1. เกษตรเลือก "ໂປຣໄຟລ໌ເກະຕົກ"</li> <li>2. ระบบแสดงประวัติการขายและแสดงค่าสถิติการขายของเกษตรกร</li> </ol>
Postcondition	เกษตรกรสามารถดูข้อมูลการขายที่ผ่านมาได้

ตารางที่ 3.2.2.9 กรณีใช้งาน (Use Case) Add product (Agriculture)

Use Case Name	UC09: Add product (Agriculture)
Primary Actor	Farmer
Goal	เกษตรกรสามารถเพิ่มสินค้าใหม่ในระบบได้
Precondition	เกษตรกรต้องเข้าสู่ระบบ
Main Flow	<ol style="list-style-type: none"> <li>1. เกษตรกรเลือก "เพิ่มสินค้า"</li> <li>2. กรอกข้อมูลสินค้า</li> <li>3. อัปโหลดรูปภาพ</li> <li>4. กดบันทึก</li> </ol>
Postcondition	สินค้าถูกเพิ่มในระบบและพร้อมสำหรับการขาย

ตารางที่ 3.2.2.10 กรณีใช้งาน (Use Case) Community Farmer

Use Case Name	UC11: Community Farmer
Primary Actor	Customer, Farmer
Goal	ผู้ใช้สามารถแลกเปลี่ยนข้อมูลหรือพูดคุยกันในชุมชน
Precondition	ผู้ใช้ต้องเข้าสู่ระบบ
Main Flow	<ol style="list-style-type: none"> <li>1. เกษตรกรเลือก "ชุมชนเกษตรกร"</li> <li>2. โพสต์หรือแสดงความคิดเห็น</li> <li>3. ตอบโต้กับผู้ใช้อื่น</li> </ol>
Postcondition	ผู้ใช้มีปฏิสัมพันธ์และแบ่งปันข้อมูลในชุมชน และเปลี่ยนความรู้ทางการเกษตร

### 3.3 การดำเนินงาน

#### 3.3.1 วิเคราะห์ความต้องการ

ในการเริ่มต้นโครงการ ผู้พัฒนาได้ทำการสัมภาษณ์เกษตรกรในพื้นที่ชุมพรเพื่อเก็บรวบรวมข้อมูลเกี่ยวกับปัญหาและความต้องการในการขายสินค้าเกษตร โดยตามถึงปัญหาและวิธีชีวิตของเกษตรกร ซึ่งผลการสัมภาษณ์พบว่าเกษตรกรส่วนใหญ่ประสบปัญหาคล้ายคลึงกันในหลายด้าน โดยสามารถสรุปได้ดังนี้

-ปัญหาที่พบจากการสัมภาษณ์เกษตรกร:

- ไม่สามารถเข้าถึงตลาดออนไลน์ได้ เนื่องจากขาดความรู้ทางเทคโนโลยี
- พิ่งพาพ่อค้าคนกลาง ซึ่งทำให้ราคาที่ขายได้ต่ำกว่าราคากลาง
- ไม่มีระบบสำหรับบริหารจัดการคำสั่งซื้อหรือสินค้าคงคลัง
- การโปรโมตสินค้าทางโซเชียลมีเดียยังไม่เป็นระบบ และไม่สามารถตอบคำถามลูกค้าได้อย่างรวดเร็ว

-ความต้องการหลักจากเกษตรกร:

- ระบบที่สามารถขายสินค้าเกษตรได้ง่าย ๆ ด้วยตนเอง
- พัฒนาระบบจัดการคำสั่งซื้อที่ดูง่าย ไม่ซับซ้อน
- สามารถแก้ไขรายละเอียดสินค้าได่อง เช่น ราคา รูปภาพ ปริมาณคงเหลือ
- เว็บไซต์ต้องใช้งานง่าย รองรับมือถือ และเป็นภาษาไทย
- ต้องการระบบแสดงประวัติคำสั่งซื้อหรือการแสดงเจ้งเตือนเมื่อมีลูกค้าสั่งสินค้า

จากการวิเคราะห์ข้อมูลดังกล่าว ทีมผู้พัฒนาจึงสรุปความต้องการระบบออกเป็น 2 กลุ่มหลัก ได้แก่:

-ความต้องการของผู้ขาย (เกษตรกร):

- ลงทะเบียนเป็นผู้ขาย
- เพิ่ม/แก้ไข/ลบรายการสินค้า
- ดูคำสั่งซื้อของลูกค้าแบบเรียลไทม์
- ดูสถิติยอดขายของสินค้า
- แก้ไขข้อมูลร้านค้าและที่อยู่ได่อง

-ความต้องการของผู้ซื้อ (ลูกค้าทั่วไป):

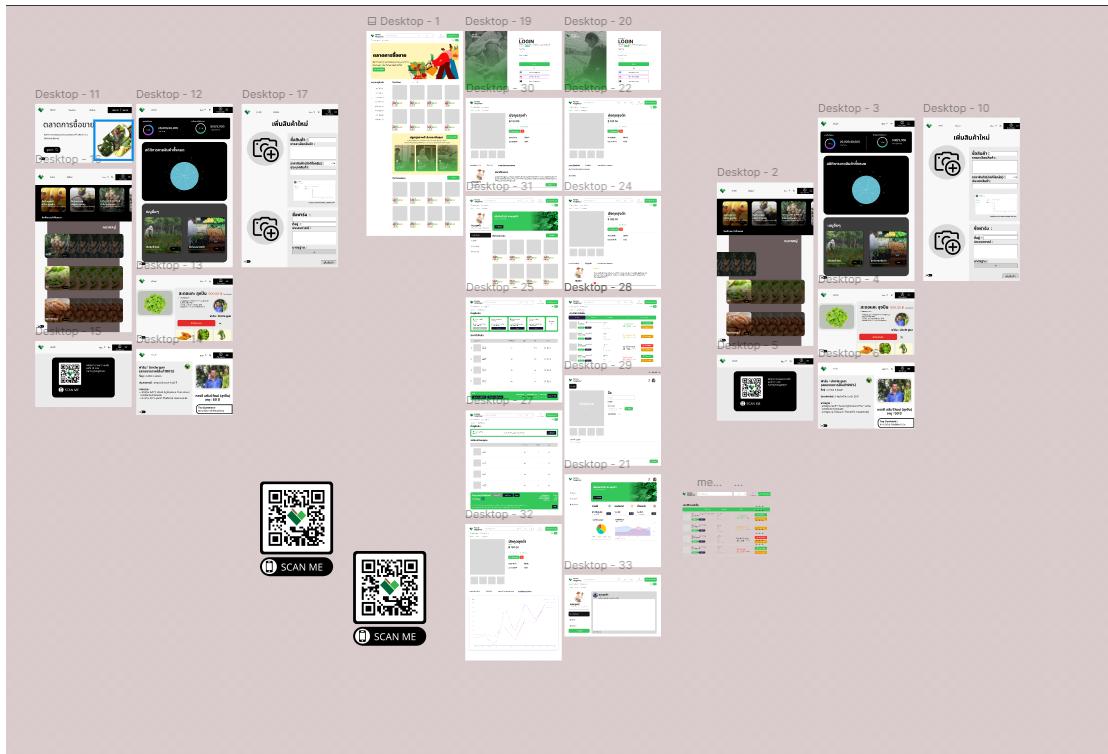
- สมัครสมาชิกและเข้าสู่ระบบ

- เลือกชุมชนค้า ค้นหา และกรองตามหมวดหมู่
- เพิ่มสินค้าในตะกร้าและยืนยันคำสั่งซื้อ
- เขียนรีวิวสินค้าได้ภายหลังการสั่งซื้อ
- ดูสถานะคำสั่งซื้อและประวัติการสั่งซื้อ

จากข้อมูลข้างต้นจะถูกใช้เป็นพื้นฐานในการออกแบบระบบ ทั้งในด้านโครงสร้างฐานข้อมูล พัฒนาการทำงาน และประสบการณ์ผู้ใช้ (UX) เพื่อให้ระบบตอบโจทย์ผู้ใช้งานเป้าหมายอย่างแท้จริง

### 3.3.2 ออกแบบระบบ

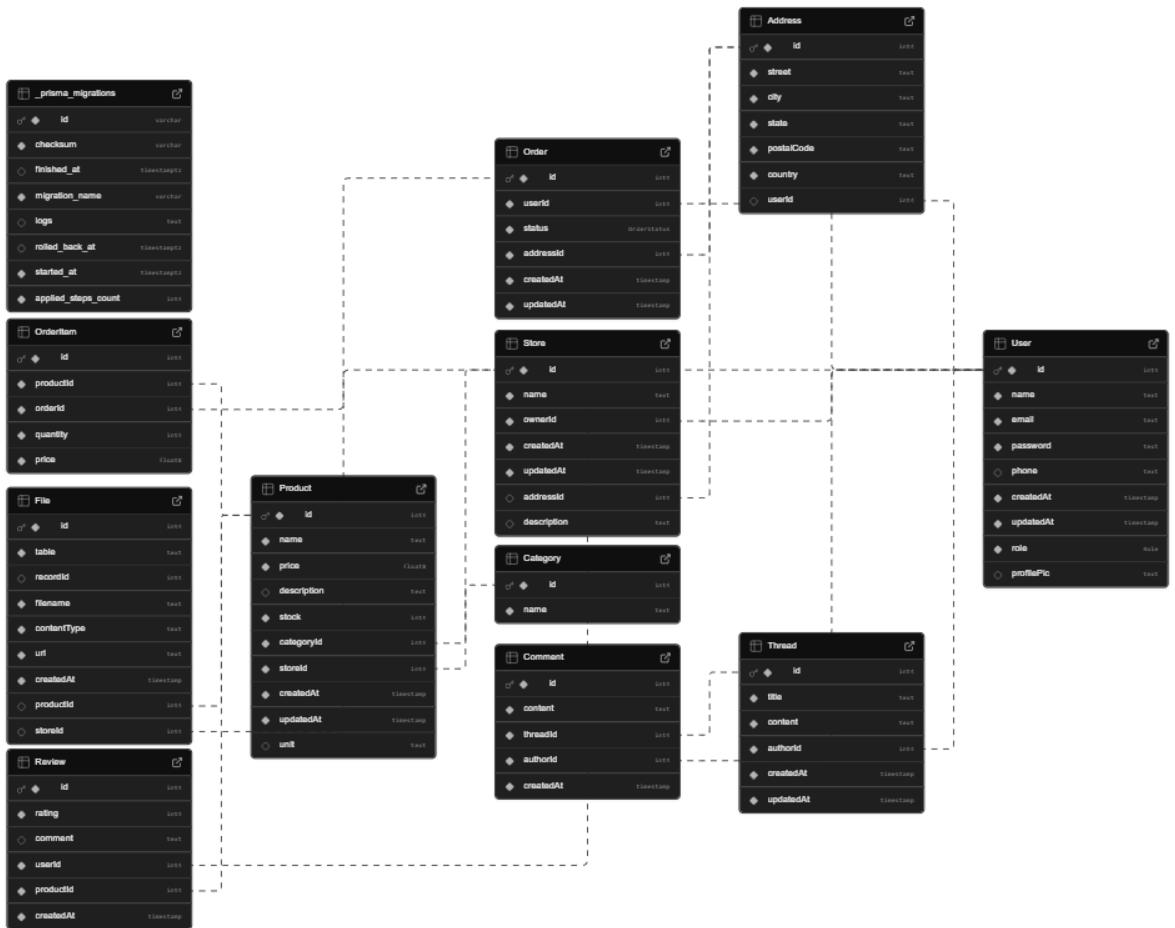
ออกแบบโครงสร้างฐานข้อมูลด้วย Prisma โดยใช้ PostgreSQL สำหรับจัดเก็บข้อมูลทั้งหมด ออกแบบ UX/UI ด้วยแนวทางที่ใช้งานง่าย รองรับผู้ใช้หลากหลายระดับ โดยใช้ Next.js เป็น Framework หลักในการพัฒนาเว็บไซต์ และแยกโครงสร้าง Backend, Frontend และ API Layer อย่างชัดเจน



ภาพที่ 3.3.2 แสดงการออกแบบ UX/UI ของระบบ

### 3.3.2.1 ออกแบบฐานข้อมูล

ระบบถูกออกแบบโดยใช้ Prisma ORM เชื่อมต่อกับ PostgreSQL ซึ่งมี Entity หลักๆ ทั้งหมด 13 ตาราง และ Enum 2 ค่า โดยแต่ละ Entity มีความสัมพันธ์กันตามความเหมาะสมของระบบ e-commerce ที่รองรับทั้งลูกค้า (Customer) และเกษตรกร (Farmer)



ภาพที่ 3.3.2.1 แสดง Er diagram ของระบบ

-Entity: User

หน้าที่:

เป็นข้อมูลของผู้ใช้ (ทั้งลูกค้าและผู้ขาย)

มี Role แยกเป็น CUSTOMER, SELLER, และ ADMIN

ความสัมพันธ์:

addresses: ที่อยู่หลายรายการ (one-to-many)

orders: ออเดอร์ที่ผู้ใช้นั้นสั่ง (one-to-many)

reviews: รีวิวที่ผู้ใช้นั้นเขียน (one-to-many)

comments, threads: ใช้ในระบบ community

store: ถ้า user เป็น SELLER จะมีร้านค้า 1 ร้าน (one-to-one)

-Entity: Store

หน้าที่:

ใช้เก็บข้อมูลร้านค้าสำหรับผู้ขาย (seller)

ความสัมพันธ์:

owner: เชื่อมกับ User (one-to-one)

products: สินค้าในร้าน (one-to-many)

images: ไฟล์รูปภาพร้าน (one-to-many ผ่าน model File)

address: ที่อยู่ของร้าน (optional one-to-one ผ่าน Address)

-Entity: File

หน้าที่:

ใช้เก็บไฟล์ต่าง ๆ เช่น รูปภาพสินค้า, ร้านค้า ๆ ฯ

ความสัมพันธ์:

เชื่อมกับ Product และ Store (optional one-to-many)

table และ recordId ใช้ระบุว่าไฟล์นี้ผูกกับ table ใดแบบ dynamic

-Entity: Product

หน้าที่:

ใช้เก็บข้อมูลสินค้า

ความสัมพันธ์:

category: หมวดหมู่สินค้า (many-to-one)

store: ร้านค้าที่ขายสินค้า (many-to-one)

images: ไฟล์รูปภาพ (one-to-many ผ่าน File)

orderItems: รายการออเดอร์ที่ส่งสินค้านี้ (many-to-many ผ่าน OrderItem)

reviews: รีวิวของสินค้านี้ (one-to-many)

- Entity: Category

หน้าที่:

หมวดหมู่สินค้า เช่น ผัก, ผลไม้, ข้าว ฯลฯ

ความสัมพันธ์:

เชื่อมกับสินค้า (one-to-many)

-Entity: Order

หน้าที่:

ใช้เก็บคำสั่งซื้อของลูกค้า

ความสัมพันธ์:

user: ใครเป็นคนสั่ง (many-to-one)

address: ที่อยู่ในการจัดส่ง (many-to-one)

orderItems: รายการสินค้าในคำสั่งซื้อนี้ (one-to-many)

-Entity: OrderItem

หน้าที่:

รายการสินค้าแต่ละชิ้นในออเดอร์หนึ่ง ๆ

ความสัมพันธ์:

order: อญี่ปุ่นออเดอร์ได (many-to-one)

product: สินค้าได (many-to-one)

-Entity: Address

หน้าที่:

ที่อยู่สำหรับจัดส่งหรือที่อยู่ร้านค้า

ความสัมพันธ์:

user: เจ้าของที่อยู่นี้ (optional many-to-one)

orders: ออเดอร์ที่ใช้ที่อยู่นี้ (one-to-many)

store: ถ้าใช้กับร้านค้า จะเป็น one-to-one

-Entity: Review

หน้าที่:

รีวิวสินค้าที่เขียนโดยผู้ใช้

ความสัมพันธ์:

user: ใครเขียนรีวิว (many-to-one)

product: รีวิวของสินค้าได (many-to-one)

-Entity: Thread

หน้าที่:

ใช้ในระบบ community (เช่น กระทู้คำถาม/แลกเปลี่ยน)

ความสัมพันธ์:

author: ผู้เขียนกระทู้ (many-to-one)

comments: คอมเม้นต์ทั้งหมดในกระทู้นี้ (one-to-many)

- Entity: Comment

หน้าที่:

ความคิดเห็นในกระทู้ (thread)

ความสัมพันธ์:

author: ผู้เขียนคอมเมนต์ (many-to-one)

thread: คอมเมนต์อยู่ในกระทู้ใด (many-to-one)

- Enum: Role

กำหนดบทบาทของผู้ใช้:

CUSTOMER: ลูกค้าธรรมดา

SELLER: ผู้ขาย

ADMIN: แอดมินระบบ

- Enum: OrderStatus

สถานะของคำสั่งซื้อ:

PENDING: รอการยืนยัน

PROCESSING: กำลังจัดเตรียม

SHIPPED: จัดส่งแล้ว

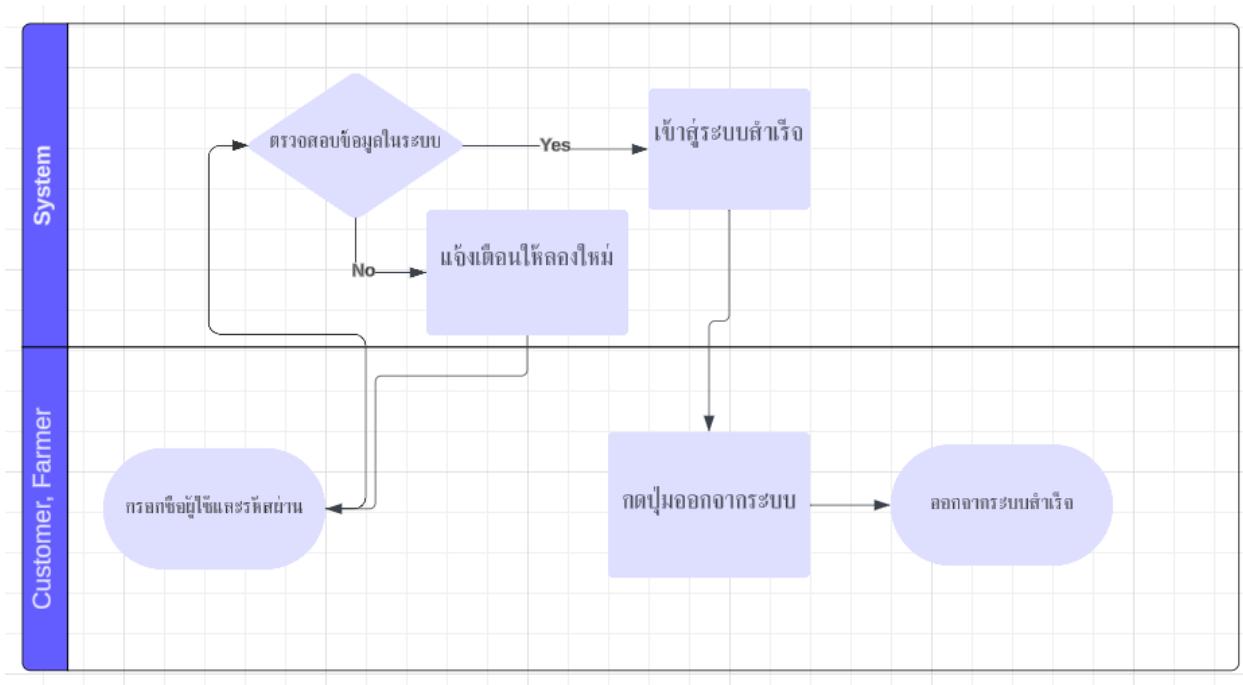
DELIVERED: ส่งถึงแล้ว

CANCELLED: ยกเลิกแล้ว

### 3.3.2.2 การออกแบบกระบวนการทำงานของระบบ

จากการวิเคราะห์การทำงานของระบบ Use Case Description ตามกรณีต่างๆ จึงสามารถนำการวิเคราะห์ที่นั้นมาออกแบบกระบวนการทำงานเป็นภาพรวมของแต่ละกรณีได้ ดังนี้

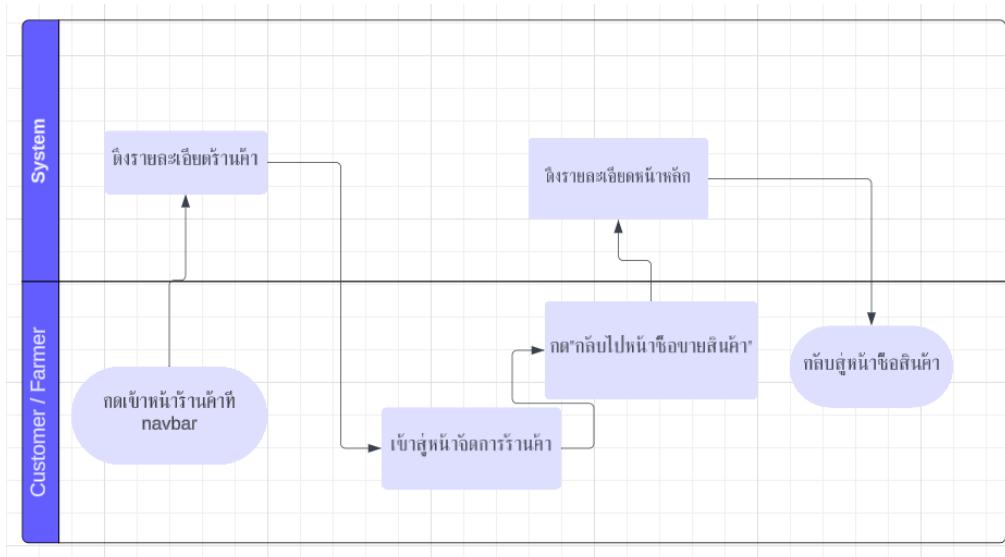
- UC01: Log in - Log out



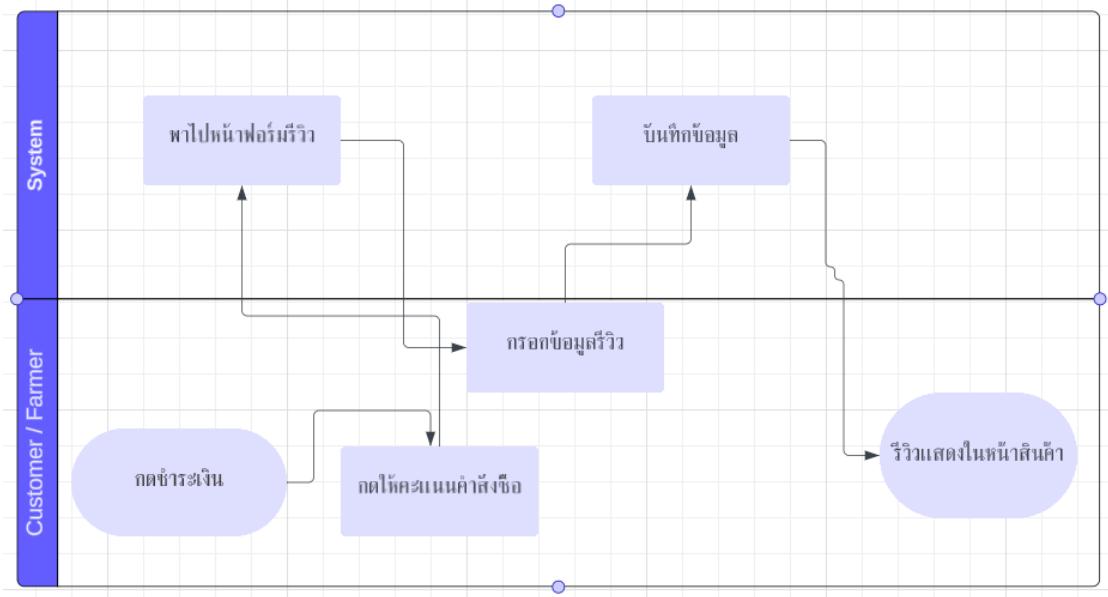
ภาพที่ 3.3.2.2.1 แสดงถึงการทำงาน Log in - Log out

- UC02: Switch Role

ภาพที่ 3.3.2.2.2 แสดงถึงการทำงาน Switch Role

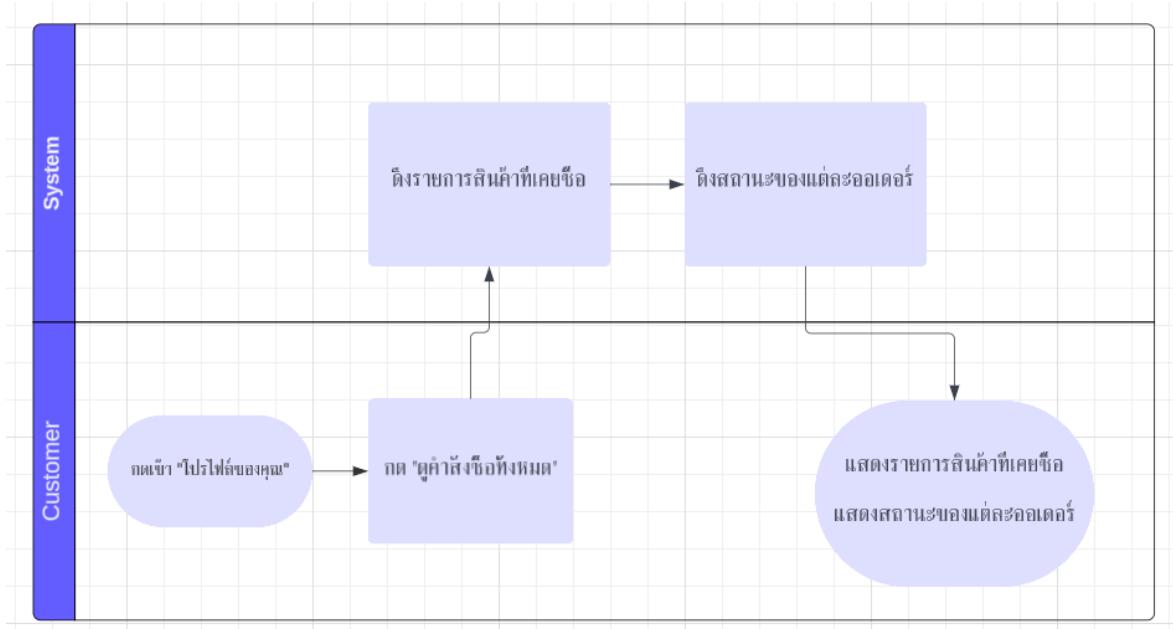


- UC03: Review - Comment



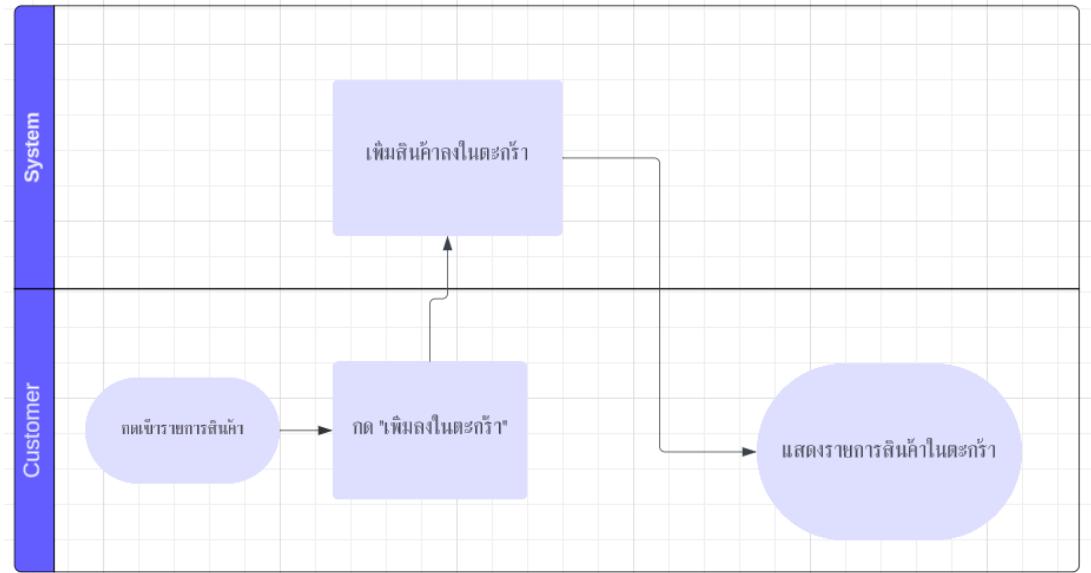
ภาพที่ 3.3.2.2.3 แสดงถึงการทำงาน Review - Comment

- UC04: Check Buy History



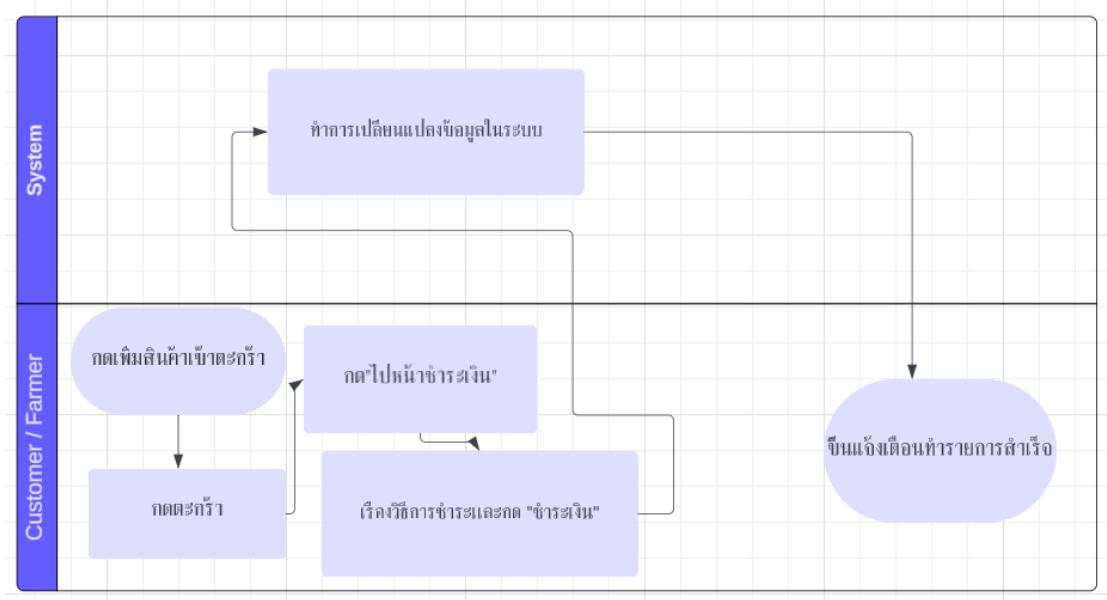
ภาพที่ 3.3.2.2.4 แสดงถึงการทำงาน Check Buy History

- UC05: Add to Basket



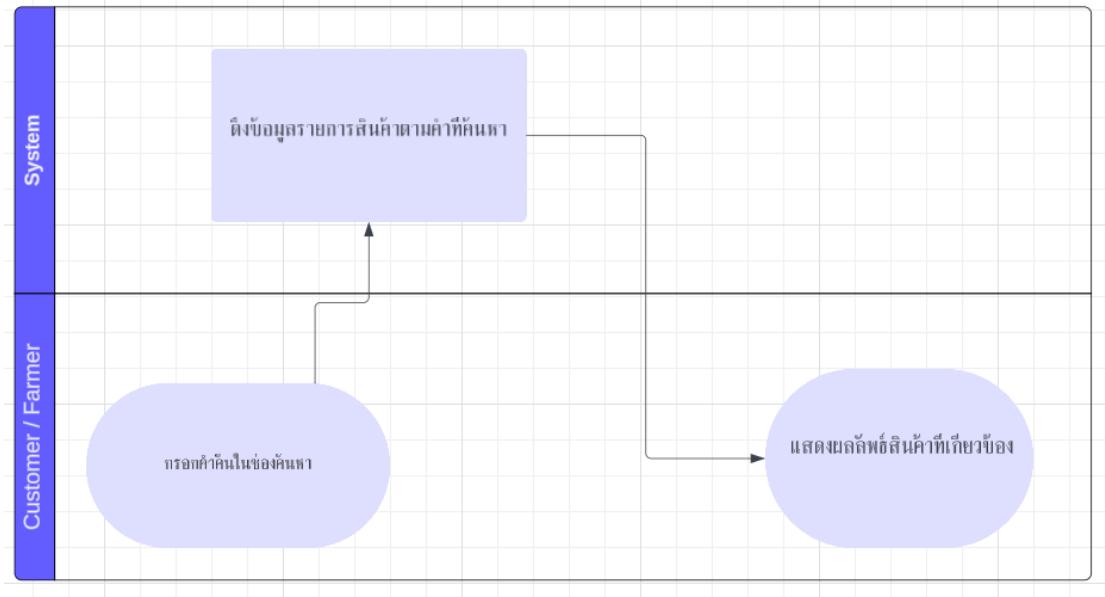
ภาพที่ 3.3.2.2.5 แสดงถึงการทำงาน Add to Basket

- UC06: Check Out Product in Basket



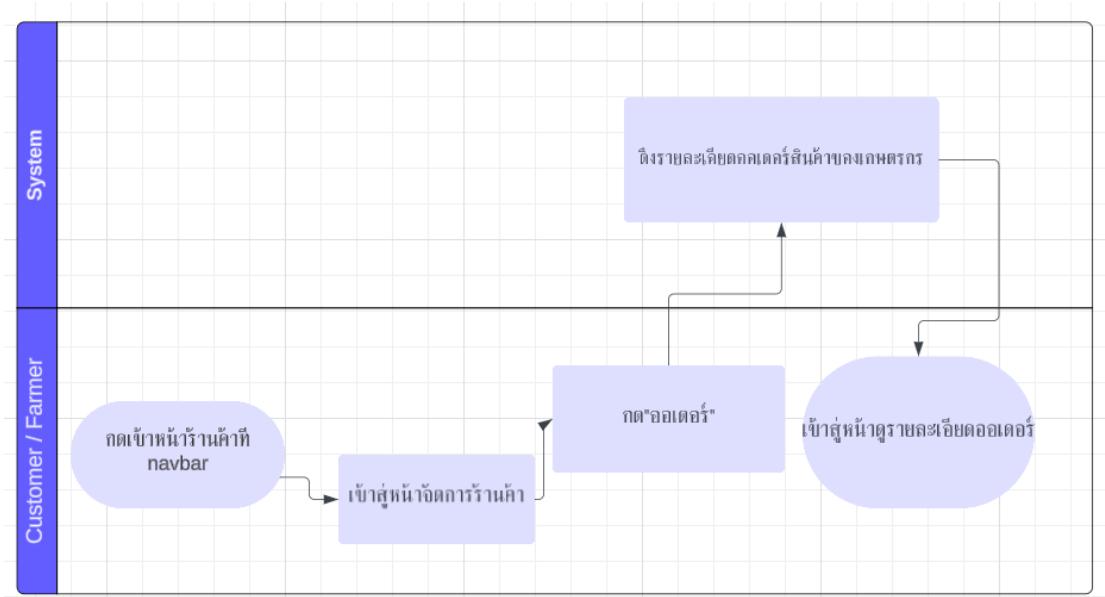
ภาพที่ 3.3.2.2.6 แสดงถึงการทำงาน Check Out Product in Basket

- UC07: Search Item



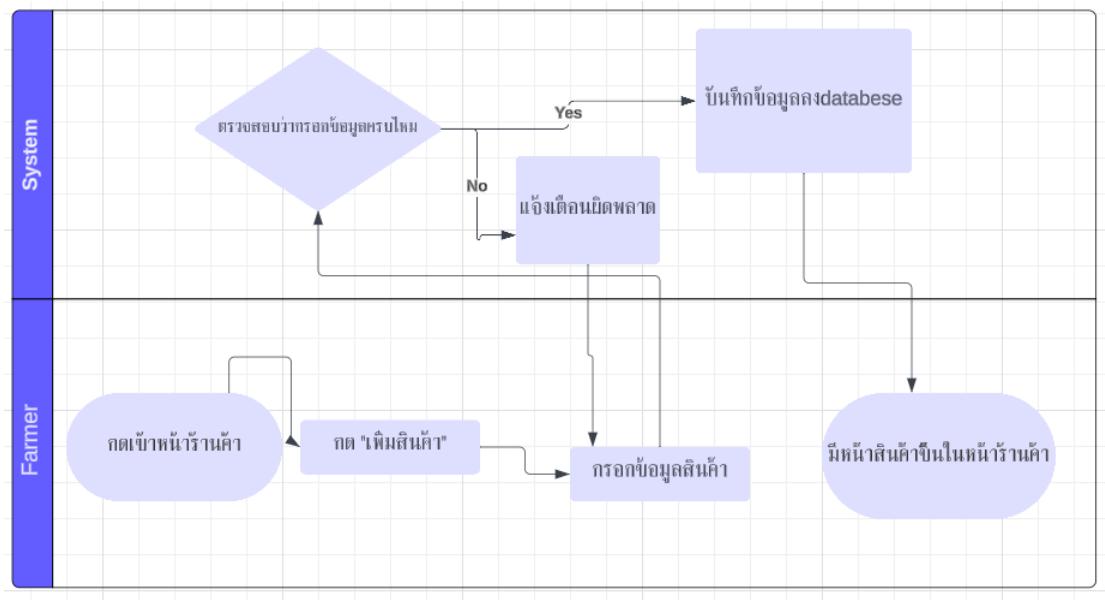
ภาพที่ 3.3.2.2.7 แสดงถึงการทำงาน Search Item

-UC08: Check Farmer History



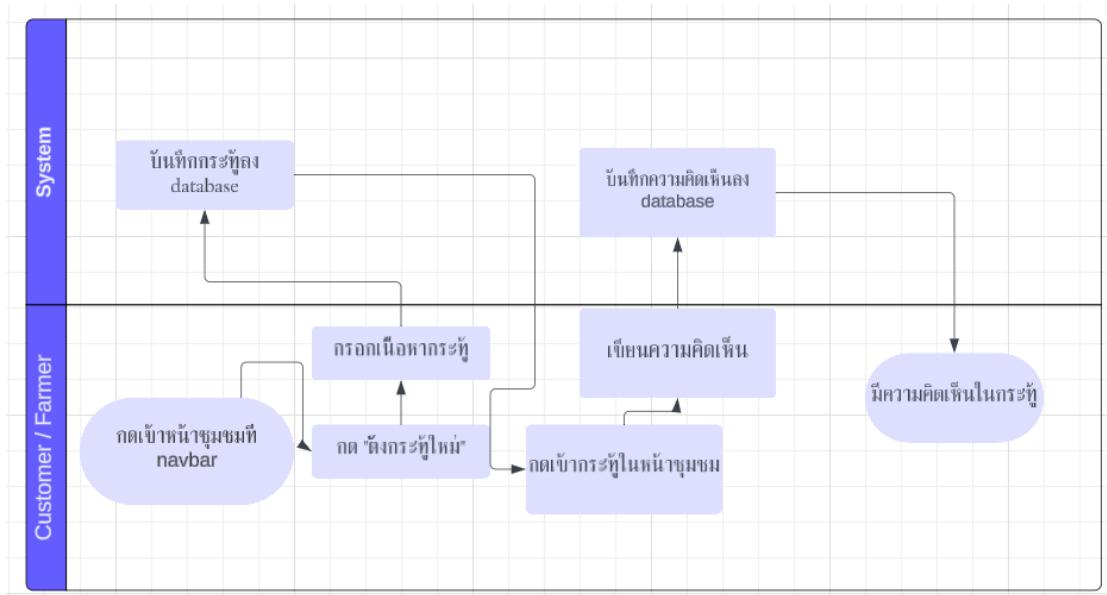
ภาพที่ 3.3.2.2.8 แสดงถึงการทำงาน Check Farmer History

- UC09: Add Product (Agriculture)



ภาพที่ 3.3.2.2.9 แสดงถึงการทำงาน Add Product (Agriculture)

- UC10: Community Farmer



ภาพที่ 3.3.2.2.10 แสดงถึงการทำงาน Community Farmer

### 3.3.3 พัฒนาระบบ

เริ่มพัฒนาระบบโดยใช้ Next.js ร่วมกับ TypeScript ในฝั่ง Frontend และ Backend พัฒนา API เชื่อมต่อกับฐานข้อมูลผ่าน Prisma และจัดทำระบบต่าง ๆ เช่น การลงทะเบียนผู้ใช้ การจัดการสินค้า ระบบติดตามสินค้า การสั่งซื้อ และระบบรีวิว

#### 3.3.3.1 การตั้งค่า Environment ของโปรเจกต์

เพื่อให้การพัฒนาเว็บไซต์ e-commerce สินค้าเกษตรเป็นไปอย่างราบรื่นและสามารถทำงานร่วมกันในที่มีได้อย่างมีประสิทธิภาพ จำเป็นต้องมีการตั้งค่าสภาพแวดล้อม (Environment) ที่เหมาะสมสำหรับการพัฒนา ซึ่งประกอบด้วยเครื่องมือและซอฟต์แวร์หลัก ดังนี้

-เครื่องมือพัฒนา

- IDE Visual Studio Code (VS Code)
- Runtime Node.js v18+
- Package Manager pnpm หรือ npm
- Version Control Git (ใช้ GitHub สำหรับเก็บ source code)
- Terminal Tool Git Bash / Terminal / zsh
- Database Client PostgreSQL และ Prisma Studio

-การติดตั้งและตั้งค่าเบื้องต้น

ขั้นตอนการตั้งค่าสภาพแวดล้อม (Local Development):

1.ติดตั้ง Node.js

- ดาวน์โหลดและติดตั้ง Node.js เวอร์ชัน 18 ขึ้นไป จาก <https://nodejs.org>

2.ติดตั้ง pnpm หรือ npm (ตามที่เลือกใช้)

- หากใช้ pnpm:

```
npm install -g pnpm
```

3.ติดตั้ง PostgreSQL

- ดาวน์โหลดจาก <https://www.postgresql.org/download/>

ตั้งค่าฐานข้อมูล ecommerce\_agriculture และผู้ใช้ที่มีสิทธิ์เข้ามายังต่อ

#### 4.Clone โปรเจกต์จาก GitHub

- git clone https://github.com/Smuggin/Farmly-FinalProject.git
- cd /Farmly-FinalProject

#### 5.ติดตั้ง dependencies

- pnpm install

#### 6.สร้างไฟล์ .env และกำหนดค่าดังนี้

```
POSTGRES_URL="postgres://postgres.ohntsosxzgjbgtaxkknb:y5hz7KXEExxyy67@aws-0-us-east-1.pooler.supabase.com:6543/postgres?sslmode=require&supa-base-pooler.x"
POSTGRES_PRISMA_URL="postgres://postgres.ohntsosxzgjbgtaxkknb:y5hz7KXEExxyy67@aws-0-us-east-1.pooler.supabase.com:6543/postgres?sslmode=require&supa-base-pooler.x"
SUPABASE_URL="https://ohntsosxzgjbgtaxkknb.supabase.co"
NEXT_PUBLIC_SUPABASE_URL="https://ohntsosxzgjbgtaxkknb.supabase.co"
POSTGRES_URL_NON_POOLING="postgres://postgres.ohntsosxzgjbgtaxkknb:y5hz7KXEExxyy67@aws-0-us-east-1.pooler.supabase.com:5432/postgres?sslmode=require"
SUPABASE_JWT_SECRET="t0GOVblyt096y3Hjdlfx4du1528uj1TGohJ0ClIAf3qcONoDU271s+s7Kj1vplEMVkhKu2+F13RHx4LMu5thigI0q=-"
POSTGRES_USER="postgres"
POSTGRES_PASSWORD="y5hz7KXEExxyy67w"
POSTGRES_DATABASE="postgres"
SUPABASE_SERVICE_ROLE_KEY="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im9odHNuc294emdqYmdhdHhra25iIiwcw9sZSI6ImPub24iLCJpYXQiOjE3NDMyNjM5OTUsImV4ccPOSTGRES_HOST="db.ohntsosxzgjbgtaxkknb.supabase.co"
SUPABASE_ANON_KEY="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im9odHNuc294emdqYmdhdHhra25iIiwcw9sZSI6ImPub24iLCJpYXQiOjE3NDMyNjM5OTUsImV4ccCI6MjA1Dgzt0DATABASE_URL="postgresql://postgres.ohntsosxzgjbgtaxkknb:turk0813443211@aws-0-us-east-1.pooler.supabase.com:6543/postgres?pgbouncer=true"
DIRECT_URL="postgresql://postgres.ohntsosxzgjbgtaxkknb:turk0813443211@aws-0-us-east-1.pooler.supabase.com:5432/postgres"
NEXT_PUBLIC_API_URL="http://localhost:3000"
GOOGLE_CLIENT_ID="20886571892-rm141of956jveeq4grtr4umeqa4fb3g.apps.googleusercontent.com"
GOOGLE_CLIENT_SECRET="GOCSPY-91o-99okBe3TFXw65h08Edk11F3"
FACEBOOK_CLIENT_ID="577611961510009"
FACEBOOK_CLIENT_SECRET="c6a39c5d83593463601937bac1753b91"
TWITTER_CLIENT_ID="TVJ0k5mWEVUmQcVt15m6H16Mpja0"
TWITTER_CLIENT_SECRET="m51cgm8Pjrb_t0Le14ndlYvmtEpsHmpHZjFQn1OcqVspJ48
```

#### ภาพที่ 3.3.3.1 แสดงถึงไฟล์ .env

#### 7.เข้ามายังฐานข้อมูลและ migrate schema

- npx prisma generate
- npx prisma migrate dev --name init

#### 8.เริ่มต้นเชิร์ฟเวอร์สำหรับพัฒนา

- pnpm dev

### 3.3.3.2 การพัฒนา Frontend

Frontend พัฒนาด้วย Next.js (App Router) โดยแบ่ง Component ตามหน้าต่างๆ เช่น หน้า Login, Register, Home, Product Detail, Profile, และ Community โดยใช้ TypeScript เพื่อเพิ่มความปลอดภัยในการพัฒนา

- มีการใช้ Layout component เช่น NavBar, MiniNavBar, ProductSection
- ใช้ Tailwind CSS ในการออกแบบ UI ให้สวยงาม สอดคล้องกับ UX ของผู้ใช้งาน
- ใช้ NextAuth.js สำหรับจัดการระบบ Authentication รองรับ OAuth (Google, Facebook, Twitter)

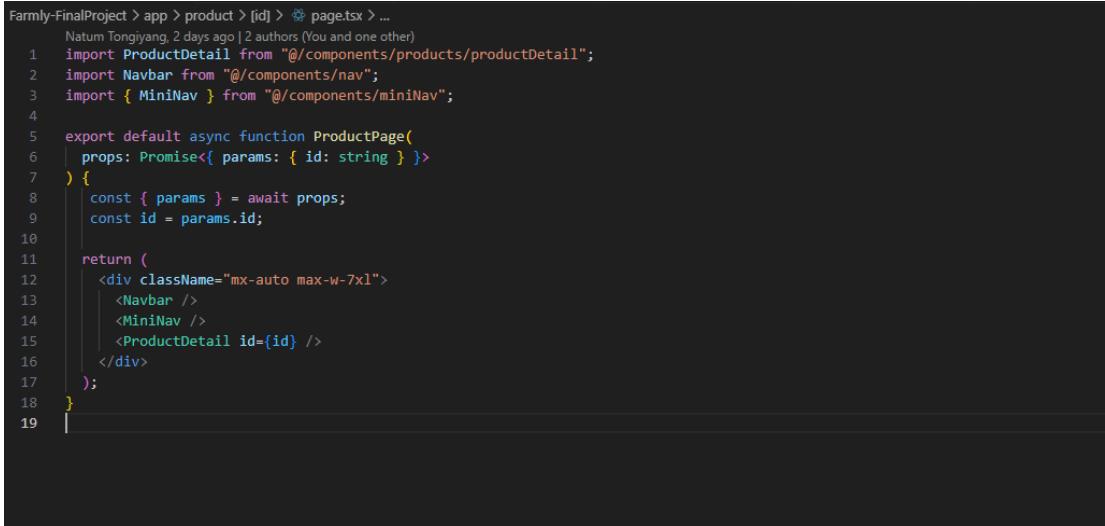
### 3.3.3.3 การพัฒนา Backend

Backend ใช้ Next.js API Routes ร่วมกับ Prisma เพื่อจัดการกับฐานข้อมูล โดยออกแบบ API สำหรับการใช้งานต่าง ๆ เช่น

- /api/auth/ สำหรับการเข้าสู่ระบบและออกจากระบบ
- /api/products/ สำหรับเพิ่ม แก้ไข ลบ และดึงข้อมูลสินค้า
- /api/orders/ สำหรับการจัดการคำสั่งซื้อ
- /api/reviews/ สำหรับรีวิวสินค้า
- /api/community/ สำหรับกระทู้และความคิดเห็น

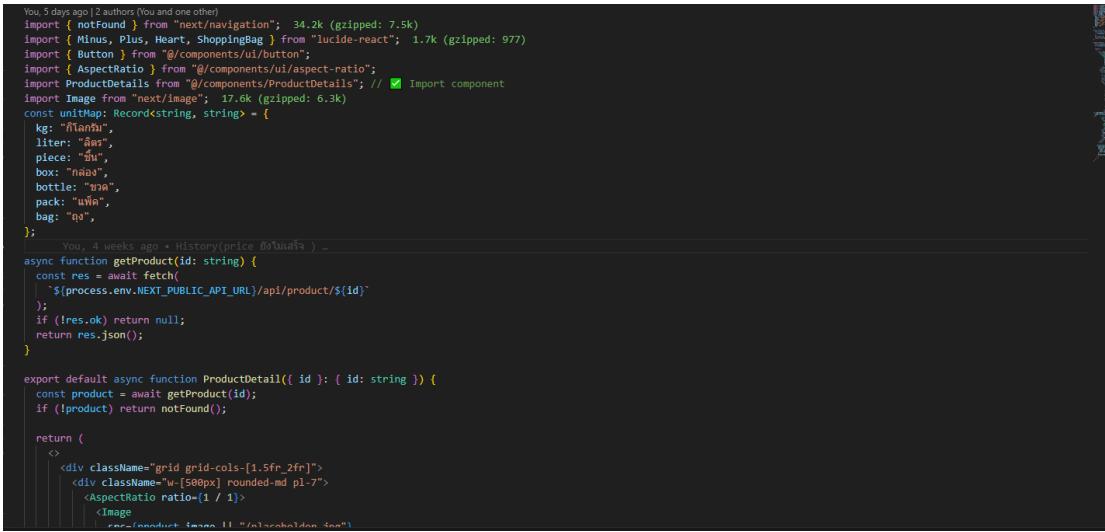
### 3.3.3.4 การออกแบบโครงสร้าง Component

- หน้าเว็บถูกพัฒนาแบบแยกส่วน (Component-based) เช่น Navbar, MiniNav, MiniMenus, ProductSection, BannerSection, และ ProductDetails
- แต่ละ Component มีหน้าที่รับ prop และแสดงผลตามข้อมูลที่ได้จากฐานข้อมูลหรือ API



```
Farmly-FinalProject > app > product > [id] > page.tsx > ...
Natum Tongiyang, 2 days ago | 2 authors (You and one other)
1 import ProductDetail from "@/components/products/productDetail";
2 import Navbar from "@/components/nav";
3 import { MiniNav } from "@/components/mininav";
4
5 export default async function ProductPage(
6   props: Promise<{ params: { id: string } } >
7 ) {
8   const { params } = await props;
9   const id = params.id;
10
11   return (
12     <div className="mx-auto max-w-7xl">
13       <Navbar />
14       <MiniNav />
15       <ProductDetail id={id} />
16     </div>
17   );
18 }
|
```

ภาพที่ 3.3.3.4.1 ตัวอย่างการเอา Component มาใช้งานโดย ส่ง id ที่เป็นของ product ไปให้ด้วย



```
You, 5 days ago | 2 authors (You and one other)
import { notFound } from "next/navigation"; 34.2k (gzipped: 7.5k)
import { Minus, Plus, Heart, ShoppingBag } from "lucide-react"; 1.7k (gzipped: 977)
import { Button } from "@components/ui/button";
import { AspectRatio } from "@components/ui/aspect-ratio";
import ProductDetails from "@components/ProductDetails"; // ✅ Import component
import Image from "next/image"; 17.6k (gzipped: 6.3k)
const unitMap: Record<string, string> = {
  kg: "กิโลกรัม",
  liter: "ลิตร",
  piece: "ชิ้น",
  box: "กล่อง",
  bottle: "ขวด",
  pack: "แพ็ค",
  bag: "ถุง",
};
// You, 4 weeks ago + History(price ถูกปรับเพิ่ม)
async function getProduct(id: string) {
  const res = await fetch(
    `${process.env.NEXT_PUBLIC_API_URL}/api/product/${id}`
  );
  if (!res.ok) return null;
  return res.json();
}

export default async function ProductDetail({ id }: { id: string }) {
  const product = await getProduct(id);
  if (!product) return notFound();

  return (
    <>
      <div className="grid grid-cols-[1.5fr_2fr]">
        <div className="w-[500px] rounded-md pl-7">
          <AspectRatio ratio={1 / 1}>
            <Image
              src={product.image || "placeholder.png"}>
          </AspectRatio>
        </div>
        <div>
```

ภาพที่ 3.3.3.4.2 ตัวอย่าง Component ที่ถูกเรียกไปใช้

### 3.3.3.5 การดึงข้อมูลด้วย Prisma และ API Route

- ใช้ Prisma ใน server function เพื่อดึงข้อมูล เช่น category และ product จากฐานข้อมูล

- หน้า product detail ใช้ fetch จาก API route ที่สร้างขึ้นภายใต้

/api/product/[id]

- มีการสร้าง API ในรูปแบบ RESTful เช่น

GET /api/product/[id], PUT /api/product/[id]

- ใช้ Prisma ใน API เพื่ออ่านและแก้ไขข้อมูลสินค้า

- มีการจัดการ Error และ Response ด้วย NextResponse

```
import { NextResponse } from "next/server"; 71.2k (gzipped: 22.2k)
import { PrismaClient } from "@prisma/client"; 41.7k (gzipped: 15.7k)

const prisma = new PrismaClient();

export async function GET(req: Request) {
  const { pathname } = new URL(req.url);
  const id = pathname.split("/").pop();

  if (!id) {
    return NextResponse.json({ error: "Invalid product ID" }, { status: 400 });
  }

  const product = await prisma.product.findUnique({
    where: { id: Number(id) },
    include: {
      category: true,
      store: true
    },
  });
  if (!product) {
    return NextResponse.json({ error: "Product not found" }, { status: 404 });
  }

  return NextResponse.json(product);
}

export async function PUT(
  req: Request,
  { params }: { params: { id: string } }
) {
  const data = await req.json();

  const updatedProduct = await prisma.product.update({
    where: { id: Number(params.id) },
    data
  });

  return NextResponse.json(updatedProduct);
}
```

ภาพที่ 3.3.3.5 ตัวอย่าง API ที่ใช้เพื่อ fetch ข้อมูลและการใช้ RESTful

### 3.3.3.6 การจัดการ Session และ Authentication

- ใช้ NextAuth ในการจัดการระบบ Login ทั้งแบบ OAuth (Google, Facebook) และ Credentials
- กำหนด JWT strategy ให้ Session เก็บไว้ใน client
- สร้าง middleware ตรวจสอบสิทธิ์การเข้าถึงหน้า Profile, Order, และหน้าสำหรับผู้ขาย

```
import NextAuth, { NextAuthOptions } from "next-auth"; 358.8k (gzipped: 96.9k)
import CredentialsProvider from "next-auth/providers/credentials"; 530 (gzipped: 327)
import GoogleProvider from "next-auth/providers/google"; // ✅ Add this 757 (gzipped: 478)
import FacebookProvider from "next-auth/providers/facebook"; // ✅ Add this 977 (gzipped: 564)
import TwitterProvider from "next-auth/providers/twitter"; // ✅ Add this 1.6k (gzipped: 787)
import { PrismaClient } from "@prisma/client"; 41.7k (gzipped: 15.7k)
import bcrypt from "bcrypt";

const prisma = new PrismaClient();

export const authOptions: NextAuthOptions = [
  providers: [
    CredentialsProvider({
      name: "Credentials",
      credentials: {
        email: { label: "Email", type: "text" },
        password: { label: "Password", type: "password" },
      },
      async authorize(credentials) {
        if (!credentials?.email || !credentials?.password) return null;

        const user = await prisma.user.findUnique({
          where: { email: credentials.email },
        });

        if (!user) return null;

        const isValid = await bcrypt.compare(credentials.password, user.password);
        if (!isValid) return null;

        return {
          id: user.id.toString(),
          name: user.name,
          email: user.email,
        },
      },
    }, // ✅ Google login
    GoogleProvider(),
  ],
]
```

ภาพที่ 3.3.3.6.1 ตัวอย่างการทำ Credentials สำหรับ Authentication

```
session: {
  strategy: "jwt",
},
callbacks: {
  async jwt({ token, user }) {
    if (user) {
      token.id = user.id;
      token.email = user.email;
    }

    if (user?.email) {
      const dbUser = await prisma.user.findUnique({
        where: { email: user.email },
        include: {
          store: true, // assuming `store` is a relation on the `user`
        }
      });
    }

    if (dbUser) {
      token.user_id = dbUser.id; // ✅ real DB user id
    } else {
      token.user_id = null;
    }
    return token;
  },
  async session({ session, token }) {
    if (session.user) {
      session.user.id = token.id as string;
      session.user.email = token.email as string;
      session.user.user_id = token.user_id as string; // ✅ real DB user id
    }
    return session;
  },
}
```

ภาพที่ 3.3.3.6.2 ตัวอย่างการกำหนด Session

```

const prisma = new PrismaClient();

export async function GET() {
  const session = await getServerSession(authOptions);
  console.log("Session", session)
  if (!session || !session.user?.email) {
    console.log("No session found")
    return NextResponse.json({ error: "Unauthorized" }, { status: 401 });
  }

  const user = await prisma.user.findUnique({
    where: { email: session.user.email },
  });

  if (!user) {
    return NextResponse.json({ error: "User not found" }, { status: 404 });
  }
}

```

ภาพที่ 3.3.3.6.3 ตัวอย่างการเรียกใช้ session เพื่อดึงข้อมูล user

### 3.3.4 ทดสอบระบบ

ดำเนินการทดสอบระบบแต่ละส่วนแยกตามฟังก์ชัน เช่น ทดสอบการลงทะเบียน การสั่งซื้อ และการแสดงสินค้า เพื่อให้มั่นใจว่าไม่มีข้อผิดพลาดในการทำงาน พร้อมทั้งทดสอบการใช้งานโดยรวมเพื่อให้ได้ประสบการณ์ผู้ใช้ที่ดีที่สุด โดยเราได้ทำการทดสอบด้วยการทดสอบด้วยตนเอง และทดสอบด้วยการนำไปให้ผู้ที่มีประสบการณ์ด้านการพัฒนาเว็บไซต์ ได้ลอง Demo ใช้งานระบบบนเครื่องของผู้ทดสอบในฐานะ “ผู้ใช้ทั่วไป” โดยไม่แนะนำฟังก์ชันล่วงหน้า ใช้เพื่อสอบถามประสบการณ์การใช้งานเพื่อนำมาปรับปรุงต่อไป

## 3.4 ประเด็นที่น่าสนใจและสิ่งที่ท้าทาย

### 3.4.1 ประเด็นที่น่าสนใจ

ในส่วนนี้สามารถอธิบายถึงประเด็นที่สำคัญหรือที่มีความน่าสนใจเกี่ยวกับการพัฒนาระบบแพลตฟอร์มซึ่งขยายสินค้าเกษตรออนไลน์ เช่น:

#### 3.4.1.1 การพัฒนาระบบที่เปิดโอกาสให้ผู้ใช้งานสามารถสลับบทบาทระหว่างผู้ซื้อและผู้ขายได้

ระบบไม่ได้จำกัดบทบาทของผู้ใช้งานให้เป็นเพียงผู้ซื้อสินค้าเท่านั้น แต่เปิดโอกาสให้ผู้ใช้ที่ลงทะเบียนแล้วสามารถสมัครเป็นผู้ขายหรือเกษตรกรได้ด้วย ทำให้บัญชีผู้ใช้มีความยืดหยุ่น

สามารถใช้งานได้หลากหลาย และตอบโจทย์กลุ่มเป้าหมายที่เป็นทั้งผู้บริโภคและผู้ผลิตสินค้าเกษตรในรายเดียวกัน

โดยระบบจะเปลี่ยนสิทธิ์การเข้าถึงเมนูต่าง ๆ เช่น การเพิ่มสินค้า การดูคำสั่งซื้อจากลูกค้า หรือการจัดการร้านค้า ซึ่งเป็นแนวคิดที่สอดคล้องกับการใช้งานจริง

### 3.4.1.2 การใช้เทคโนโลยี Next.js (App Router) ร่วมกับ Prisma และ PostgreSQL

ระบบพัฒนาด้วย Next.js ที่ใช้ App Router โครงสร้างใหม่ล่าสุด ร่วมกับ Prisma ORM และฐานข้อมูล PostgreSQL ซึ่งเป็นการใช้เทคโนโลยีที่ทันสมัย มีความสามารถในการพัฒนา Full-stack ได้ในเฟรมเวิร์กเดียว และเหมาะสมกับการพัฒนาแอปพลิเคชันเชิงโมดูลาร์

### 3.4.1.3 การใช้เทคโนโลยีสมัยใหม่ที่ผสม Frontend และ Backend อย่างไร้อย่างต่อ

ระบบนี้พัฒนาด้วย Next.js (App Router) ซึ่งเป็นแนวทางใหม่ที่รวมการเขียนฝั่งเซิร์ฟเวอร์ (Server Component) และฝั่งไคลเอนต์ (Client Component) ไว้ในเฟรมเวิร์กเดียว ช่วยให้พัฒนา Full-stack Application ได้ภายในโค้ดเดียว ทำให้การจัดการ UI และ Data Fetching มีความยืดหยุ่นและประสิทธิภาพสูง

ใช้ Prisma ORM ในการติดต่อฐานข้อมูล PostgreSQL ซึ่งช่วยให้เขียนโค้ดจัดการข้อมูลได้สะดวก และสามารถเชื่อมโยงโมเดลที่มีความซับซ้อน เช่น Order, User, Store, Product, Review, Address ได้อย่างชัดเจนผ่าน Prisma schema

## 3.4.2 สิ่งที่ท้าทาย

### 3.4.2.1 การจัดการความสัมพันธ์ระหว่างข้อมูลในฐานข้อมูลที่มีความซับซ้อน

โครงสร้างของระบบมีความซับซ้อนพอสมควร เช่น ผู้ใช้หนึ่งคนสามารถมีที่อยู่ได้หลายรายการ (One-to-Many), คำสั่งซื้อหนึ่งรายการอ้างอิงสินค้าหลายรายการผ่าน OrderItem (Many-to-Many), หรือแต่ละสินค้าต้องอ้างอิงถึงหมวดหมู่และร้านค้า ซึ่งความสัมพันธ์เหล่านี้ต้องการความเข้าใจอย่างลึกซึ้งในการออกแบบ Prisma Schema รวมถึงการเขียน Query เพื่อให้ดึงข้อมูลที่ถูกต้องครบถ้วน โดยไม่ทำให้ระบบช้าหรือซับซ้อนเกินจำเป็น

### 3.4.2.2 การออกแบบ API และฟังก์ชันการทำงานที่รองรับหลายบทบาทผู้ใช้งาน

ฟังก์ชันบางส่วน เช่น การเพิ่มสินค้า, การดูคำสั่งซื้อ, หรือการแสดงข้อมูลส่วนตัว ต้องแยกสิทธิ์และเงื่อนไขตามบทบาทของผู้ใช้งาน (เช่น ลูกค้า/เกษตรกร) ซึ่งต้องออกแบบให้มีความยืดหยุ่นและปลอดภัยในระดับที่เพียงพอ

### 3.4.2.3 การออกแบบสิทธิ์การใช้งานและการควบคุมการเข้าถึง (Access Control)

การพัฒนาให้ผู้ใช้สามารถสั่งซื้อและจ่ายได้ด้วยตัวเอง ต้องมีการจัดการสิทธิ์อย่างชัดเจนในระดับ Routing, Component และ API เช่น การห้ามไม่ให้ผู้ใช้ทั่วไปเข้าถึงหน้าเพิ่มสินค้า, การให้เฉพาะเจ้าของร้านดูคำสั่งซื้อได้ หรือการตรวจสอบ session และ role ของผู้ใช้ทุกครั้งที่มีการร้องขอข้อมูล

ความท้าทายคือการทำให้ระบบมีความปลอดภัยโดยไม่ลดตอนประสบการณ์การใช้งาน และต้องสามารถจัดการข้อผิดพลาดได้หากมีการพยายามเข้าถึงหน้าที่ไม่มีสิทธิ์

## 3.5 ผลลัพธ์ที่คาดหวัง

### 3.5.1 ความสะดวกและความเร็วในการซื้อขายสินค้าเกษตรออนไลน์

หนึ่งในผลลัพธ์ที่สำคัญที่สุดคือการทำให้ การซื้อขายสินค้าเกษตร กลายเป็นเรื่องที่ง่ายและสะดวกยิ่งขึ้นผ่านช่องทางออนไลน์ ผู้บริโภคสามารถค้นหาสินค้าได้อย่างรวดเร็ว ไม่ว่าจะเป็นผลผลิตสดหรือสินค้าผลิตภัณฑ์แปรรูป โดยใช้ ฟังก์ชันการค้นหาและการกรองสินค้า ที่ช่วยให้ผู้ใช้สามารถหาสินค้าได้ตามต้องการ เช่น เลือกตามประเภทสินค้า ราคา หรือสินค้าที่มีการรับรองคุณภาพ (เช่น GAP หรือ GMP) การใช้ Server-Side Rendering (SSR) ในการโหลดหน้าเว็บจะช่วยเพิ่มความเร็วในการเข้าถึงข้อมูล โดยเฉพาะการแสดงข้อมูลสินค้าและรีวิวจากผู้ใช้งาน ส่งผลให้ผู้บริโภคไม่ต้องรอนานในการตัดสินใจซื้อสินค้า ทำให้การซื้อขายมีความเร็วสูงขึ้น ซึ่งจะช่วยกระตุ้นการซื้อและเพิ่มยอดขายให้กับเกษตรกร

### 3.5.2 การสร้างความเชื่อมั่นให้กับเกษตรกรและผู้บริโภค

ระบบที่มีการออกแบบเพื่อให้มี ความโปร่งใส ในการทำธุกรรมและการติดตาม สถานะคำสั่งซื้อจะช่วยเพิ่มความ เชื่อมั่น ให้กับทั้งเกษตรกรและผู้บริโภค การที่ผู้บริโภคสามารถเห็นข้อมูลสินค้าทั้งหมดอย่างชัดเจน เช่น ชื่อสินค้า รายละเอียดสินค้า การรับรองคุณภาพ หรือรีวิวจาก

ผู้ใช้งาน จะช่วยสร้างความมั่นใจในการตัดสินใจซื้อสินค้าเกษตรออนไลน์ นอกจากนี้ ระบบจะช่วยเกษตรกรในการติดตามสถานะการจัดส่งและประวัติการขาย ทำให้พวกรเขามั่นใจว่าการขายสินค้าออนไลน์ของพวกรเขาก็จะได้รับการดูแลอย่างดีและโปร่งใส

สำหรับผู้บริโภค ความสามารถในการ ตรวจสอบสถานะคำสั่งซื้อ แบบเรียลไทม์ ผ่านระบบที่รองรับการแจ้งเตือน เช่น SMS หรืออีเมล ก็จะช่วยเพิ่มความสบายใจและความมั่นใจในการซื้อสินค้าเกษตรออนไลน์อีกทางหนึ่ง ซึ่งจะทำให้ผู้บริโภค มีประสบการณ์ที่ดีและสร้างความเชื่อมั่นในการเลือกซื้อสินค้าจากแพลตฟอร์มนี้

### 3.5.3 การเพิ่มการเข้าถึงตลาดสำหรับเกษตรกร

การใช้เทคโนโลยีที่ทันสมัยจะช่วยให้ เกษตรกร สามารถขยายการเข้าถึงตลาดและขายสินค้าของตนได้ง่ายขึ้น โดยเฉพาะการขายสินค้าผ่านช่องทางออนไลน์ที่ไม่มีข้อจำกัดทางด้านพื้นที่หรือเวลา การมี เว็บไซต์ที่เข้าถึงง่าย และรองรับการใช้งานบนมือถือ (Responsive Design) จะทำให้เกษตรกรสามารถเข้าถึงกลุ่มผู้บริโภคที่หลากหลาย รวมถึงผู้บริโภคในพื้นที่ห่างไกลที่ไม่สะดวกไปหาซื้อสินค้าที่ตลาด

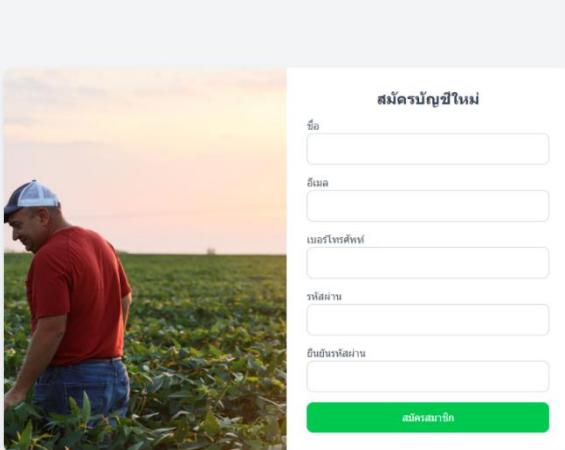
นอกจากนี้ การรองรับ การชำระเงินผ่านระบบ Mobile Banking ยังเป็นการขยายช่องทางการชำระเงินให้สะดวกสบายยิ่งขึ้น และช่วยให้เกษตรกรสามารถรับเงินได้อย่างรวดเร็วและปลอดภัย ซึ่งจะทำให้พวกรสามารถจัดการการเงินได้ง่ายขึ้น และมีช่องทางการขายที่กว้างขึ้น รวมถึงการแนะนำสินค้า (Product Recommendation) ที่จะช่วยให้สินค้าเกษตรของพวกรเขามีโอกาสสูงในการขายให้กับผู้บริโภคที่สนใจสินค้าที่คล้ายคลึงกัน

## บทที่ 4

### ผลการดำเนินงาน

#### 4.1 หน้าจอหลักของระบบ

##### 4.1.1 UC01: Log in – Log out



```

    name: "form",
    password: "",
    confirmPassword: ""
  );
}

const [passwordError, setPasswordError] = useState("");
const [apiResponse, setApiResponse] = useState("");

const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {
  setFormData({ ...formData, [e.target.name]: e.target.value });
};

const handleSubmit = async (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault();
}

// Check if passwords match
if (formData.password !== formData.confirmPassword) {
  setPasswordError("กําเนดูไสสູ່ຫມາວາຫຼຸງ");
  return;
}

// You 3 months ago + loginRegisterHistory
setPasswordError("");

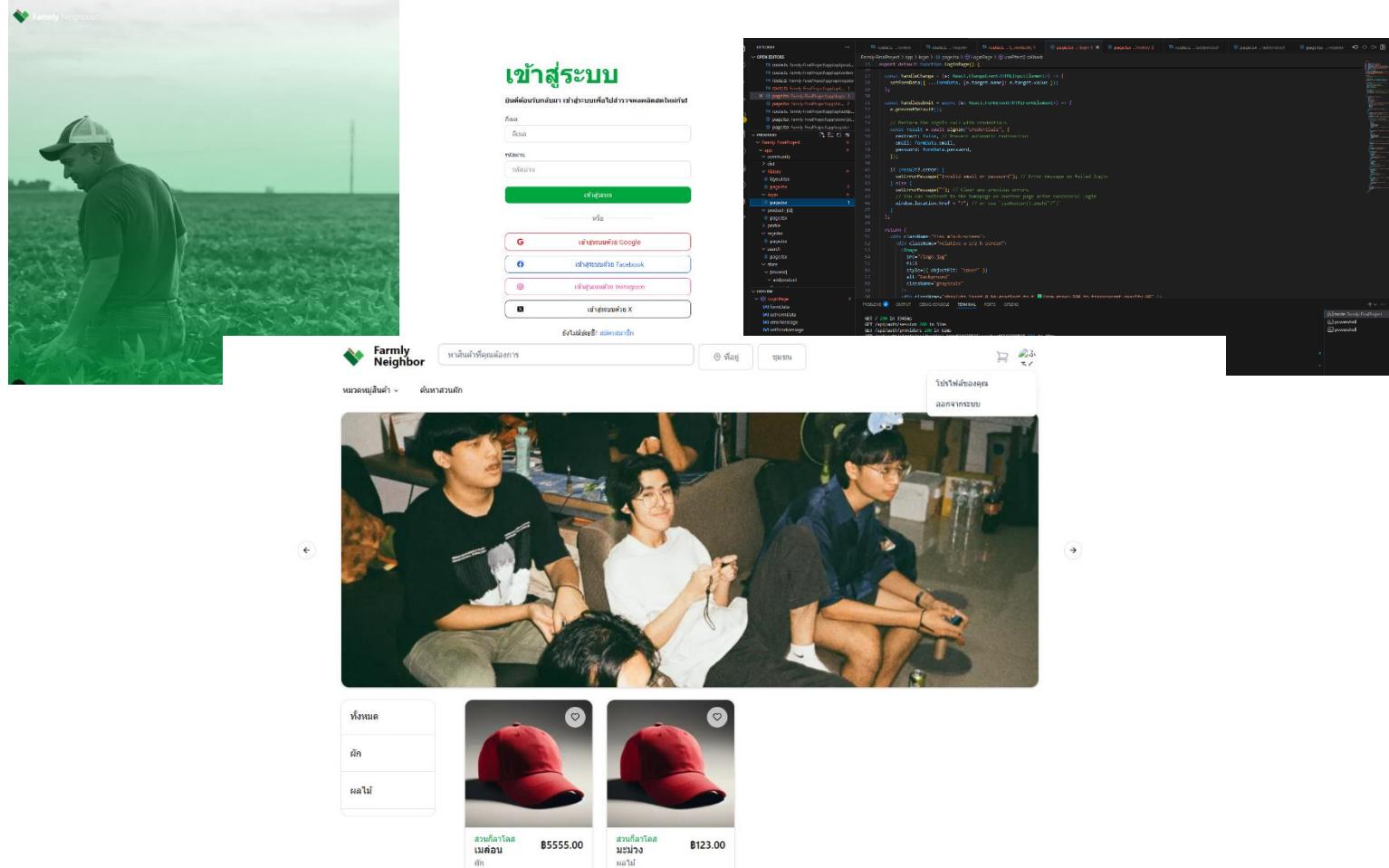
try {
  const res = await fetch("/api/register", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({
      name: formData.name,
      email: formData.email,
      phone: formData.phone,
      password: formData.password,
    })
  });
}

```

下方显示了命令行界面的输出：

```

$ npx ts-node index.ts
[{"id": 1, "name": "John Doe", "email": "john.doe@example.com", "password": "password123", "phone": "0987654321", "status": "Active", "created_at": "2023-01-01T10:00:00Z", "updated_at": "2023-01-01T10:00:00Z"}, {"id": 2, "name": "Jane Smith", "email": "jane.smith@example.com", "password": "password456", "phone": "0987654322", "status": "Active", "created_at": "2023-01-02T10:00:00Z", "updated_at": "2023-01-02T10:00:00Z"}, {"id": 3, "name": "Mike Johnson", "email": "mike.johnson@example.com", "password": "password789", "phone": "0987654323", "status": "Active", "created_at": "2023-01-03T10:00:00Z", "updated_at": "2023-01-03T10:00:00Z"}, {"id": 4, "name": "Sarah Williams", "email": "sarah.williams@example.com", "password": "password123", "phone": "0987654324", "status": "Active", "created_at": "2023-01-04T10:00:00Z", "updated_at": "2023-01-04T10:00:00Z"}, {"id": 5, "name": "David Miller", "email": "david.miller@example.com", "password": "password456", "phone": "0987654325", "status": "Active", "created_at": "2023-01-05T10:00:00Z", "updated_at": "2023-01-05T10:00:00Z"}, {"id": 6, "name": "Emily Davis", "email": "emily.davis@example.com", "password": "password789", "phone": "0987654326", "status": "Active", "created_at": "2023-01-06T10:00:00Z", "updated_at": "2023-01-06T10:00:00Z"}, {"id": 7, "name": "Aaron Wilson", "email": "aaron.wilson@example.com", "password": "password123", "phone": "0987654327", "status": "Active", "created_at": "2023-01-07T10:00:00Z", "updated_at": "2023-01-07T10:00:00Z"}, {"id": 8, "name": "Bella Lewis", "email": "bella.lewis@example.com", "password": "password456", "phone": "0987654328", "status": "Active", "created_at": "2023-01-08T10:00:00Z", "updated_at": "2023-01-08T10:00:00Z"}, {"id": 9, "name": "Caleb Parker", "email": "caleb.parker@example.com", "password": "password789", "phone": "0987654329", "status": "Active", "created_at": "2023-01-09T10:00:00Z", "updated_at": "2023-01-09T10:00:00Z"}, {"id": 10, "name": "Diana Clark", "email": "diana.clark@example.com", "password": "password123", "phone": "0987654330", "status": "Active", "created_at": "2023-01-10T10:00:00Z", "updated_at": "2023-01-10T10:00:00Z"}, {"id": 11, "name": "Elijah Foster", "email": "elijah.foster@example.com", "password": "password456", "phone": "0987654331", "status": "Active", "created_at": "2023-01-11T10:00:00Z", "updated_at": "2023-01-11T10:00:00Z"}, {"id": 12, "name": "Fiona Hayes", "email": "fiona.hayes@example.com", "password": "password789", "phone": "0987654332", "status": "Active", "created_at": "2023-01-12T10:00:00Z", "updated_at": "2023-01-12T10:00:00Z"}, {"id": 13, "name": "Gavin Thompson", "email": "gavin.thompson@example.com", "password": "password123", "phone": "0987654333", "status": "Active", "created_at": "2023-01-13T10:00:00Z", "updated_at": "2023-01-13T10:00:00Z"}, {"id": 14, "name": "Hannah Martinez", "email": "hannah.martinez@example.com", "password": "password456", "phone": "0987654334", "status": "Active", "created_at": "2023-01-14T10:00:00Z", "updated_at": "2023-01-14T10:00:00Z"}, {"id": 15, "name": "Ivy Rodriguez", "email": "ivy.rodriguez@example.com", "password": "password789", "phone": "0987654335", "status": "Active", "created_at": "2023-01-15T10:00:00Z", "updated_at": "2023-01-15T10:00:00Z"}, {"id": 16, "name": "Jasmine Wilson", "email": "jasmine.wilson@example.com", "password": "password123", "phone": "0987654336", "status": "Active", "created_at": "2023-01-16T10:00:00Z", "updated_at": "2023-01-16T10:00:00Z"}, {"id": 17, "name": "Kai Thompson", "email": "kai.thompson@example.com", "password": "password456", "phone": "0987654337", "status": "Active", "created_at": "2023-01-17T10:00:00Z", "updated_at": "2023-01-17T10:00:00Z"}, {"id": 18, "name": "Liam Martinez", "email": "liam.martinez@example.com", "password": "password789", "phone": "0987654338", "status": "Active", "created_at": "2023-01-18T10:00:00Z", "updated_at": "2023-01-18T10:00:00Z"}, {"id": 19, "name": "Mia Wilson", "email": "mia.wilson@example.com", "password": "password123", "phone": "0987654339", "status": "Active", "created_at": "2023-01-19T10:00:00Z", "updated_at": "2023-01-19T10:00:00Z"}, {"id": 20, "name": "Natalie Thompson", "email": "natalie.thompson@example.com", "password": "password456", "phone": "0987654340", "status": "Active", "created_at": "2023-01-20T10:00:00Z", "updated_at": "2023-01-20T10:00:00Z"}, {"id": 21, "name": "Owen Martinez", "email": "owen.martinez@example.com", "password": "password789", "phone": "0987654341", "status": "Active", "created_at": "2023-01-21T10:00:00Z", "updated_at": "2023-01-21T10:00:00Z"}, {"id": 22, "name": "Peyton Wilson", "email": "peyton.wilson@example.com", "password": "password123", "phone": "0987654342", "status": "Active", "created_at": "2023-01-22T10:00:00Z", "updated_at": "2023-01-22T10:00:00Z"}, {"id": 23, "name": "Quinn Thompson", "email": "quinn.thompson@example.com", "password": "password456", "phone": "0987654343", "status": "Active", "created_at": "2023-01-23T10:00:00Z", "updated_at": "2023-01-23T10:00:00Z"}, {"id": 24, "name": "Riley Martinez", "email": "riley.martinez@example.com", "password": "password789", "phone": "0987654344", "status": "Active", "created_at": "2023-01-24T10:00:00Z", "updated_at": "2023-01-24T10:00:00Z"}, {"id": 25, "name": "Savannah Wilson", "email": "savannah.wilson@example.com", "password": "password123", "phone": "0987654345", "status": "Active", "created_at": "2023-01-25T10:00:00Z", "updated_at": "2023-01-25T10:00:00Z"}, {"id": 26, "name": "Tanner Thompson", "email": "tanner.thompson@example.com", "password": "password456", "phone": "0987654346", "status": "Active", "created_at": "2023-01-26T10:00:00Z", "updated_at": "2023-01-26T10:00:00Z"}, {"id": 27, "name": "Ulysses Martinez", "email": "ulysses.martinez@example.com", "password": "password789", "phone": "0987654347", "status": "Active", "created_at": "2023-01-27T10:00:00Z", "updated_at": "2023-01-27T10:00:00Z"}, {"id": 28, "name": "Vivian Wilson", "email": "vivian.wilson@example.com", "password": "password123", "phone": "0987654348", "status": "Active", "created_at": "2023-01-28T10:00:00Z", "updated_at": "2023-01-28T10:00:00Z"}, {"id": 29, "name": "Wade Thompson", "email": "wade.thompson@example.com", "password": "password456", "phone": "0987654349", "status": "Active", "created_at": "2023-01-29T10:00:00Z", "updated_at": "2023-01-29T10:00:00Z"}, {"id": 30, "name": "Xavier Martinez", "email": "xavier.martinez@example.com", "password": "password789", "phone": "0987654350", "status": "Active", "created_at": "2023-01-30T10:00:00Z", "updated_at": "2023-01-30T10:00:00Z"}, {"id": 31, "name": "Yara Wilson", "email": "yara.wilson@example.com", "password": "password123", "phone": "0987654351", "status": "Active", "created_at": "2023-01-31T10:00:00Z", "updated_at": "2023-01-31T10:00:00Z"}, {"id": 32, "name": "Zane Thompson", "email": "zane.thompson@example.com", "password": "password456", "phone": "0987654352", "status": "Active", "created_at": "2023-02-01T10:00:00Z", "updated_at": "2023-02-01T10:00:00Z"}, {"id": 33, "name": "Ava Martinez", "email": "ava.martinez@example.com", "password": "password789", "phone": "0987654353", "status": "Active", "created_at": "2023-02-02T10:00:00Z", "updated_at": "2023-02-02T10:00:00Z"}, {"id": 34, "name": "Caleb Wilson", "email": "caleb.wilson@example.com", "password": "password123", "phone": "0987654354", "status": "Active", "created_at": "2023-02-03T10:00:00Z", "updated_at": "2023-02-03T10:00:00Z"}, {"id": 35, "name": "Diana Thompson", "email": "diana.thompson@example.com", "password": "password456", "phone": "0987654355", "status": "Active", "created_at": "2023-02-04T10:00:00Z", "updated_at": "2023-02-04T10:00:00Z"}, {"id": 36, "name": "Elijah Martinez", "email": "elijah.martinez@example.com", "password": "password789", "phone": "0987654356", "status": "Active", "created_at": "2023-02-05T10:00:00Z", "updated_at": "2023-02-05T10:00:00Z"}, {"id": 37, "name": "Fiona Wilson", "email": "fiona.wilson@example.com", "password": "password123", "phone": "0987654357", "status": "Active", "created_at": "2023-02-06T10:00:00Z", "updated_at": "2023-02-06T10:00:00Z"}, {"id": 38, "name": "Gavin Thompson", "email": "gavin.thompson@example.com", "password": "password456", "phone": "0987654358", "status": "Active", "created_at": "2023-02-07T10:00:00Z", "updated_at": "2023-02-07T10:00:00Z"}, {"id": 39, "name": "Hannah Martinez", "email": "hannah.martinez@example.com", "password": "password789", "phone": "0987654359", "status": "Active", "created_at": "2023-02-08T10:00:00Z", "updated_at": "2023-02-08T10:00:00Z"}, {"id": 40, "name": "Ivy Wilson", "email": "ivy.wilson@example.com", "password": "password123", "phone": "0987654360", "status": "Active", "created_at": "2023-02-09T10:00:00Z", "updated_at": "2023-02-09T10:00:00Z"}, {"id": 41, "name": "Jasmine Thompson", "email": "jasmine.thompson@example.com", "password": "password456", "phone": "0987654361", "status": "Active", "created_at": "2023-02-10T10:00:00Z", "updated_at": "2023-02-10T10:00:00Z"}, {"id": 42, "name": "Kai Martinez", "email": "kai.martinez@example.com", "password": "password789", "phone": "0987654362", "status": "Active", "created_at": "2023-02-11T10:00:00Z", "updated_at": "2023-02-11T10:00:00Z"}, {"id": 43, "name": "Liam Wilson", "email": "liam.wilson@example.com", "password": "password123", "phone": "0987654363", "status": "Active", "created_at": "2023-02-12T10:00:00Z", "updated_at": "2023-02-12T10:00:00Z"}, {"id": 44, "name": "Mia Thompson", "email": "mia.thompson@example.com", "password": "password456", "phone": "0987654364", "status": "Active", "created_at": "2023-02-13T10:00:00Z", "updated_at": "2023-02-13T10:00:00Z"}, {"id": 45, "name": "Natalie Martinez", "email": "natalie.martinez@example.com", "password": "password789", "phone": "0987654365", "status": "Active", "created_at": "2023-02-14T10:00:00Z", "updated_at": "2023-02-14T10:00:00Z"}, {"id": 46, "name": "Owen Wilson", "email": "owen.wilson@example.com", "password": "password123", "phone": "0987654366", "status": "Active", "created_at": "2023-02-15T10:00:00Z", "updated_at": "2023-02-15T10:00:00Z"}, {"id": 47, "name": "Peyton Thompson", "email": "peyton.thompson@example.com", "password": "password456", "phone": "0987654367", "status": "Active", "created_at": "2023-02-16T10:00:00Z", "updated_at": "2023-02-16T10:00:00Z"}, {"id": 48, "name": "Quinn Martinez", "email": "quinn.martinez@example.com", "password": "password789", "phone": "0987654368", "status": "Active", "created_at": "2023-02-17T10:00:00Z", "updated_at": "2023-02-17T10:00:00Z"}, {"id": 49, "name": "Riley Wilson", "email": "riley.wilson@example.com", "password": "password123", "phone": "0987654369", "status": "Active", "created_at": "2023-02-18T10:00:00Z", "updated_at": "2023-02-18T10:00:00Z"}, {"id": 50, "name": "Savannah Thompson", "email": "savannah.thompson@example.com", "password": "password456", "phone": "0987654370", "status": "Active", "created_at": "2023-02-19T10:00:00Z", "updated_at": "2023-02-19T10:00:00Z"}, {"id": 51, "name": "Tanner Martinez", "email": "tanner.martinez@example.com", "password": "password789", "phone": "0987654371", "status": "Active", "created_at": "2023-02-20T10:00:00Z", "updated_at": "2023-02-20T10:00:00Z"}, {"id": 52, "name": "Ulysses Wilson", "email": "ulysses.wilson@example.com", "password": "password123", "phone": "0987654372", "status": "Active", "created_at": "2023-02-21T10:00:00Z", "updated_at": "2023-02-21T10:00:00Z"}, {"id": 53, "name": "Vivian Thompson", "email": "vivian.thompson@example.com", "password": "password456", "phone": "0987654373", "status": "Active", "created_at": "2023-02-22T10:00:00Z", "updated_at": "2023-02-22T10:00:00Z"}, {"id": 54, "name": "Wade Martinez", "email": "wade.martinez@example.com", "password": "password789", "phone": "0987654374", "status": "Active", "created_at": "2023-02-23T10:00:00Z", "updated_at": "2023-02-23T10:00:00Z"}, {"id": 55, "name": "Xavier Wilson", "email": "xavier.wilson@example.com", "password": "password123", "phone": "0987654375", "status": "Active", "created_at": "2023-02-24T10:00:00Z", "updated_at": "2023-02-24T10:00:00Z"}, {"id": 56, "name": "Yara Thompson", "email": "yara.thompson@example.com", "password": "password456", "phone": "0987654376", "status": "Active", "created_at": "2023-02-25T10:00:00Z", "updated_at": "2023-02-25T10:00:00Z"}, {"id": 57, "name": "Zane Martinez", "email": "zane.martinez@example.com", "password": "password789", "phone": "0987654377", "status": "Active", "created_at": "2023-02-26T10:00:00Z", "updated_at": "2023-02-26T10:00:00Z"}, {"id": 58, "name": "Ava Wilson", "email": "ava.wilson@example.com", "password": "password123", "phone": "0987654378", "status": "Active", "created_at": "2023-02-27T10:00:00Z", "updated_at": "2023-02-27T10:00:00Z"}, {"id": 59, "name": "Caleb Thompson", "email": "caleb.thompson@example.com", "password": "password456", "phone": "0987654379", "status": "Active", "created_at": "2023-02-28T10:00:00Z", "updated_at": "2023-02-28T10:00:00Z"}, {"id": 60, "name": "Diana Martinez", "email": "diana.martinez@example.com", "password": "password789", "phone": "0987654380", "status": "Active", "created_at": "2023-03-01T10:00:00Z", "updated_at": "2023-03-01T10:00:00Z"}, {"id": 61, "name": "Elijah Wilson", "email": "elijah.wilson@example.com", "password": "password123", "phone": "0987654381", "status": "Active", "created_at": "2023-03-02T10:00:00Z", "updated_at": "2023-03-02T10:00:00Z"}, {"id": 62, "name": "Fiona Thompson", "email": "fiona.thompson@example.com", "password": "password456", "phone": "0987654382", "status": "Active", "created_at": "2023-03-03T10:00:00Z", "updated_at": "2023-03-03T10:00:00Z"}, {"id": 63, "name": "Gavin Martinez", "email": "gavin.martinez@example.com", "password": "password789", "phone": "0987654383", "status": "Active", "created_at": "2023-03-04T10:00:00Z", "updated_at": "2023-03-04T10:00:00Z"}, {"id": 64, "name": "Hannah Wilson", "email": "hannah.wilson@example.com", "password": "password123", "phone": "0987654384", "status": "Active", "created_at": "2023-03-05T10:00:00Z", "updated_at": "2023-03-05T10:00:00Z"}, {"id": 65, "name": "Ivy Thompson", "email": "ivy.thompson@example.com", "password": "password456", "phone": "0987654385", "status": "Active", "created_at": "2023-03-06T10:00:00Z", "updated_at": "2023-03-06T10:00:00Z"}, {"id": 66, "name": "Jasmine Martinez", "email": "jasmine.martinez@example.com", "password": "password789", "phone": "0987654386", "status": "Active", "created_at": "2023-03-07T10:00:00Z", "updated_at": "2023-03-07T10:00:00Z"}, {"id": 67, "name": "Kai Wilson", "email": "kai.wilson@example.com", "password": "password123", "phone": "0987654387", "status": "Active", "created_at": "2023-03-08T10:00:00Z", "updated_at": "2023-03-08T10:00:00Z"}, {"id": 68, "name": "Liam Thompson", "email": "liam.thompson@example.com", "password": "password456", "phone": "0987654388", "status": "Active", "created_at": "2023-03-09T10:00:00Z", "updated_at": "2023-03-09T10:00:00Z"}, {"id": 69, "name": "Mia Martinez", "email": "mia.martinez@example.com", "password": "password789", "phone": "0987654389", "status": "Active", "created_at": "2023-03-10T10:00:00Z", "updated_at": "2023-03-10T10:00:00Z"}, {"id": 70, "name": "Natalie Wilson", "email": "natalie.wilson@example.com", "password": "password123", "phone": "0987654390", "status": "Active", "created_at": "2023-03-11T10:00:00Z", "updated_at": "2023-03-11T10:00:00Z"}, {"id": 71, "name": "Owen Thompson", "email": "owen.thompson@example.com", "password": "password456", "phone": "0987654391", "status": "Active", "created_at": "2023-03-12T10:00:00Z", "updated_at": "2023-03-12T10:00:00Z"}, {"id": 72, "name": "Peyton Martinez", "email": "peyton.martinez@example.com", "password": "password789", "phone": "0987654392", "status": "Active", "created_at": "2023-03-13T10:00:00Z", "updated_at": "2023-03-13T10:00:00Z"}, {"id": 73, "name": "Quinn Wilson", "email": "quinn.wilson@example.com", "password": "password123", "phone": "0987654393", "status": "Active", "created_at": "2023-03-14T10:00:00Z", "updated_at": "2023-03-14T10:00:00Z"}, {"id": 74, "name": "Riley Thompson", "email": "riley.thompson@example.com", "password": "password456", "phone": "0987654394", "status": "Active", "created_at": "2023-03-15T10:00:00Z", "updated_at": "2023-03-15T10:00:00Z"}, {"id": 75, "name": "Vivian Martinez", "email": "vivian.martinez@example.com", "password": "password789", "phone": "0987654395", "status": "Active", "created_at": "2023-03-16T10:00:00Z", "updated_at": "2023-03-16T10:00:00Z"}, {"id": 76, "name": "Wade Wilson", "email": "wade.wilson@example.com", "password": "password123", "phone": "0987654396", "status": "Active", "created_at": "2023-03-17T10:00:00Z", "updated_at": "2023-03-17T10:00:00Z"}, {"id": 77, "name": "Xavier Thompson", "email": "xavier.thompson@example.com", "password": "password456", "phone": "0987654397", "status": "Active", "created_at": "2023-03-18T10:00:00Z", "updated_at": "2023-03-18T10:00:00Z"}, {"id": 78, "name": "Yara Martinez", "email": "yara.martinez@example.com", "password": "password789", "phone": "0987654398", "status": "Active", "created_at": "2023-03-19T10:00:00Z", "updated_at": "2023-03-19T10:00:00Z"}, {"id": 79, "name": "Zane Wilson", "email": "zane.wilson@example.com", "password": "password123", "phone": "0987654399", "status": "Active", "created_at": "2023-03-20T10:00:00Z", "updated_at": "2023-03-20T10:00:00Z"}, {"id": 80, "name": "Ava Thompson", "email": "ava.thompson@example.com", "password": "password456", "phone": "0987654400", "status": "Active", "created_at": "2023-03-21T10:00:00Z", "updated_at": "2023-03-21T10:00:00Z"}, {"id": 81, "name": "Caleb Martinez", "email": "caleb.martinez@example.com", "password": "password789", "phone": "0987654401", "status": "Active", "created_at": "2023-03-22T10:00:00Z", "updated_at": "2023-03-22T10:00:00Z"}, {"id": 82, "name": "Diana Wilson", "email": "diana.wilson@example.com", "password": "password123", "phone": "0987654402", "status": "Active", "created_at": "2023-03-23T10:00:00Z", "updated_at": "2023-03-23T10:00:00Z"}, {"id": 83, "name": "Elijah Thompson", "email": "elijah.thompson@example.com", "password": "password456", "phone": "0987654403", "status": "Active", "created_at": "2023-03-24T10:00:00Z", "updated_at": "2023-03-24T10:00:00Z"}, {"id": 84, "name": "Fiona Martinez", "email": "fiona.martinez@example.com", "password": "password789", "phone": "0987654404", "status": "Active", "created_at": "2023-03-25T10:00:00Z", "updated_at": "2023-03-25T10:00:00Z"}, {"id": 85, "name": "Gavin Wilson", "email": "gavin.wilson@example.com", "password": "password123", "phone": "0987654405", "status": "Active", "created_at": "2023-03-26T10:00:00Z", "updated_at": "2023-03-26T10:00:00Z"}, {"id": 86, "name": "Hannah Thompson", "email": "hannah.thompson@example.com", "password": "password456", "phone": "0987654406", "status": "Active", "created_at": "2023-03-27T10:00:00Z", "updated_at": "2023-03-27T10:00:00Z"}, {"id": 87, "name": "Ivy Martinez", "email": "ivy.martinez@example.com", "password": "password789", "phone": "0987654407", "status": "Active", "created_at": "2023-03-28T10:00:00Z", "updated_at": "2023-03-28T10:00:00Z"}, {"id": 88, "name": "Jasmine Wilson", "email": "jasmine.wilson@example.com", "password": "password123", "phone": "0987654408", "status": "Active", "created_at": "2023-03-29T10:00:00Z", "updated_at": "2023-03-29T10:00:00Z"}, {"id": 89, "name": "Kai Thompson", "email": "kai.thompson@example.com", "password": "password456", "phone": "0987654409", "status": "Active", "created_at": "2023-03-30T10:00:00Z", "updated_at": "2023-03-30T10:00:00Z"}, {"id": 90, "name": "Liam Martinez", "email": "liam.martinez@example.com", "password": "password789", "phone": "0987654410", "status": "Active", "created_at": "2023-03-31T10:00:00Z", "updated_at": "2023-03-31T10:00:00Z"}, {"id": 91, "name": "Mia Wilson", "email": "mia.wilson@example.com", "password": "password123", "phone": "0987654411", "status": "Active", "created_at": "2023-04-01T10:00:00Z", "updated_at": "2023-04-01T10:00:00Z"}, {"id": 92, "name": "Natalie Thompson", "email": "natalie.thompson@example.com", "password": "password456", "phone": "0987654412", "status": "Active", "created_at": "2023-04-02T10:00:00Z", "updated_at": "2023-04-02T10:00:00Z"}, {"id": 93, "name": "Owen Martinez", "email": "owen.martinez@example.com", "password": "password789", "phone": "0987654413", "status": "Active", "created_at": "2023-04-03T10:00:00Z", "updated_at": "2023-04-03T10:00:00Z"}, {"id": 94, "name": "Peyton Wilson", "email": "peyton.wilson@example.com", "password": "password123", "phone": "0987654414", "status": "Active", "created_at": "2023-04-04T10:00:00Z", "updated_at": "2023-04-04T10:00:00Z"}, {"id": 95, "name": "Quinn Thompson", "email": "quinn.thompson@example.com", "password": "password456", "phone": "0987654415", "status": "Active", "created_at": "2023-04-05T10:00:00Z", "updated_at": "2023-04-05T10:00:00Z"}, {"id": 96, "name": "Riley Martinez", "email": "riley.martinez@example.com", "password": "password789", "phone": "0987654416", "status": "Active", "created_at": "2023-04-06T10:00:00Z", "updated_at": "2023-04-06T10:00:00Z"}, {"id": 97, "name": "Vivian Wilson", "email": "vivian.wilson@example.com", "password": "password123", "phone": "098765441
```



ภาพที่ 4.1.1.2 แสดงหน้าเข้าสู่ระบบ การออกแบบและตัวอย่าง Code

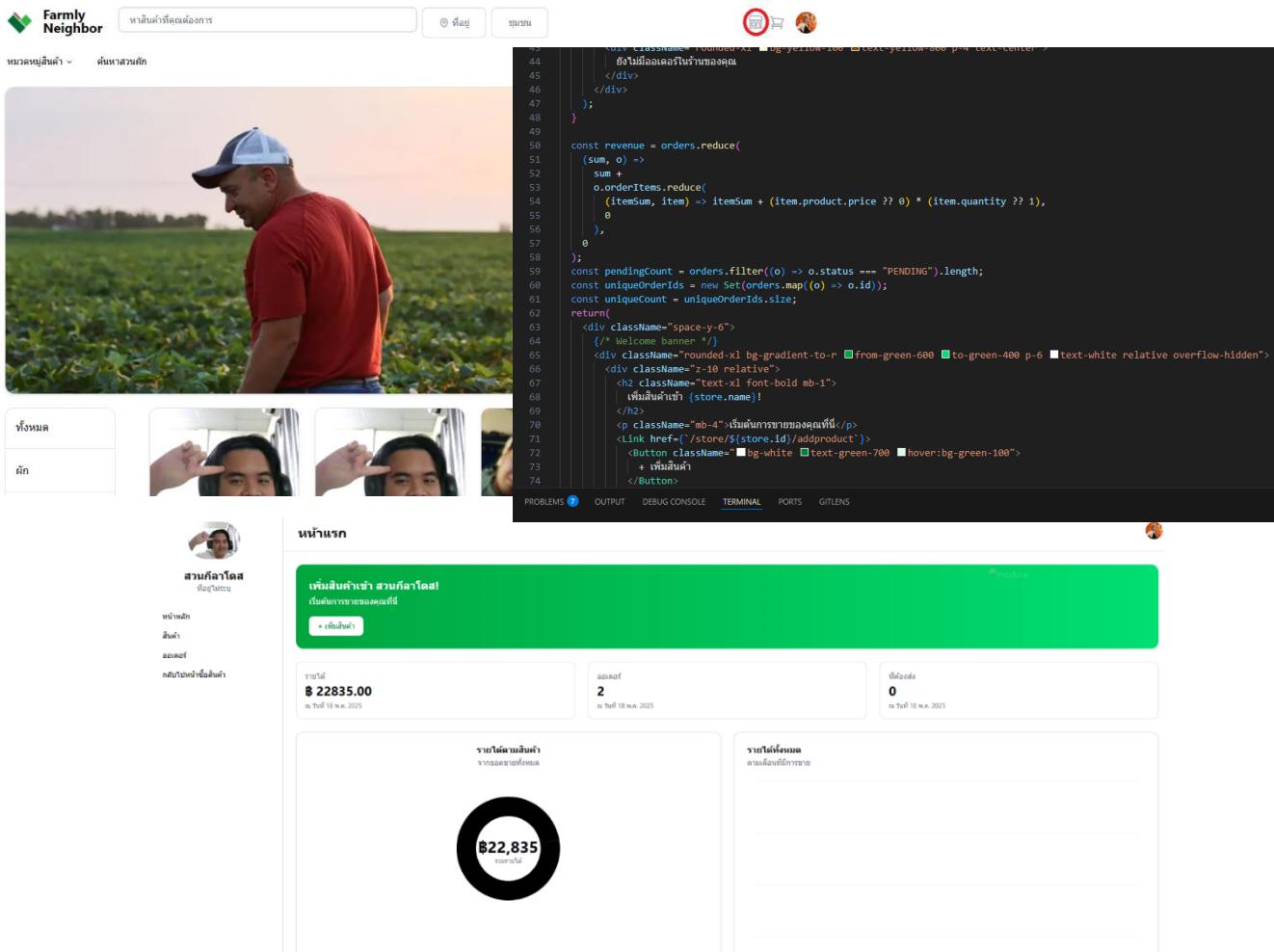
จากภาพที่ 4.1.1.2 แสดงหน้าเข้าสู่ระบบที่เราสามารถเข้าสู่ระบบจากบัญชีที่เราสร้างหรือจะเข้าสู่ระบบด้วยโซเชียลมีเดียอื่นเช่น Google หรือ Facebook ถ้าเกิดข้อผิดพลาดจะขึ้นแจ้งเตือน เมื่อเข้าสู่ระบบแล้วจะเข้าหน้าหลักซึ่งในหน้าหลักที่ miniprofile เราสามารถดูจากระบบได้

-Input: ผู้ใช้กรอกอีเมลและรหัสผ่าน หรือเลือกเข้าสู่ระบบผ่าน Google / Facebook

-Process: ระบบตรวจสอบข้อมูลบัญชีจากฐานข้อมูลหรือระบบ OAuth ของ Google/Facebook

-Output: เข้าสู่ระบบสำเร็จ: นำผู้ใช้ไปยังหน้า “หน้าหลัก” หากข้อมูลผิดพลาด: แสดงข้อความแจ้งเตือน เช่น “รหัสผ่านไม่ถูกต้อง”

#### 4.1.2 UC02: Switch Role



ภาพที่ 4.1.2 แสดงหน้าสลับระหว่างเป็นผู้ซื้อกับเป็นผู้ขายและแล清晨อย่าง Code

จากภาพ 4.1.2 แสดงหน้าสลับระหว่างเป็นผู้ซื้อกับเป็นผู้ขายซึ่งสามารถทำได้โดยการกดไอคอนตรง Navbar ในวง แล้วระบบก็จะนำพาไปหน้าจัดการร้านค้า ซึ่งถ้าเราต้องการกลับไปซื้อสินค้าก็สามารถกด กลับไปหน้าซื้อสินค้า ได้

**Input:** ผู้ใช้ที่เข้าสู่ระบบแล้ว คลิกไอคอนที่ Navbar เพื่อสลับบทบาทระหว่าง “ผู้ซื้อ” กับ “เกษตรกร”

**Process:** ระบบเปลี่ยนบทบาทของผู้ใช้และเปลี่ยนเส้นทางไปยังหน้าจัดการร้านค้า

**Output:** แสดงหน้า “ร้านค้า” ของเกษตรกร มีปุ่ม “กลับไปหน้าซื้อสินค้า” ให้เปลี่ยนกลับเป็นบทบาทผู้ซื้อ

### 4.1.3 UC03: Review – Comment

The screenshot shows a user interface for managing reviews. On the left, there's a sidebar with a green header "เพิ่มรีวิวสินค้า". Below it are sections for "คะแนน:" (Rating) set to 5ดาว (5 stars), and "ความคิดเห็น:" (Comment) containing the text "อื้ออยมาก". On the right, there's a navigation bar with tabs: "รายละเอียดสินค้า", "รีวิวสินค้า" (which is highlighted in blue), and "รายละเอียดของเกณฑ์ครอง". Below the navigation is a section titled "รีวิวจากลูกค้า" (Reviews from customers) showing a single review by "Smuggin'" with a 4-star rating. A large green button at the bottom is labeled "ลงรีวิว" (Post review).

```

], [productId]);

return (
  <div className="mt-6">
    <h3 className="font-semibold text-lg">รีวิวจากลูกค้า</h3>

    {loading ? (
      <div className="space-y-4 mt-4">
        {[1, 2, 3].map((i) => [
          <div key={i}>
            <div className="animate-pulse border p-4 rounded-md space-y-2">
              <small>Natum Tongiyang, 5 hours ago + reviews and update UX ...</small>
              <div className="h-4 bg-gray-300 rounded w-1/3" />
              <div className="h-3 bg-gray-300 rounded w-24" />
              <div className="h-3 bg-gray-300 rounded w-3/4" />
            </div>
          </div>
        ])
      </div>
    ) : reviews.length === 0 ? (
      <p className="text-gray-500 mt-4">ยังไม่มีรีวิว</p>
    ) : (
      <div className="mt-4 space-y-4">
        {reviews.map((review) => (
          <div key={review.id} className="border p-4 rounded-md">
            <p className="font-semibold">
              {review.user?.name || "ผู้ใช้งานใหม่"}
            </p>
        ))
      </div>
    )
  </div>
)
  
```

ภาพที่ 4.1.3 แสดงหน้าฟอร์มรีวิวและหน้าสินค้าที่มีรีวิวและตัวอย่าง Code

จากภาพที่ 4.1.3 แสดงหน้าฟอร์มรีวิวและหน้าสินค้าที่มีรีวิว เมื่อเราเข้าระบบ เรากำลังจะเห็นหน้าฟอร์ม การรีวิว เมื่อเรารีวิวเสร็จแล้วในหน้าสินค้าตรงรายละเอียดสินค้า “รีวิวสินค้า” ก็จะมีรีวิวขึ้นมา

Input: ผู้ใช้ที่เข้าระบบสามารถเขียนรีวิวสินค้าในหน้าสินค้า (ข้อความ + คะแนน)

Process: ระบบบันทึกรีวิวลงฐานข้อมูลและเชื่อมโยงกับสินค้านั้น

Output: รีวิวจะแสดงอยู่ในหน้ารายละเอียดของสินค้านั้นในส่วน “รีวิวสินค้า”

#### 4.1.4 UC04: Check Buy History

```

        | });
      , [status]);
    if (status === "loading") return <p>กำลังโหลด...</p>;
    if (status === "unauthenticated") return <p>กรุณาเข้าสู่ระบบก่อนจะประวัติคำสั่ง</p>;
  const filteredOrders =
    selectedTab === "ทั้งหมด" ? orders : orders.filter((order) => order.status === selectedTab);
  return (
    <div className="px-8">
      <h1 className="text-2xl font-bold mb-4">ประวัติการซื้อ</h1>
      <div className="flex gap-2 mb-4">
        {tabs.map((tab) => (
          <button
            key={tab}
            className={`px-4 py-2 rounded-lg ${(
              selectedTab === tab ? "bg-green-500 text-white" : "bg-gray-200 text-black"
            )}`}
            onClick={() => setSelectedTab(tab)}
          >
            {tab}
          </button>
        )));
      </div>
      <div className="space-y-4">
        {filteredOrders.length > 0 ? (
          filteredOrders.map((order) => <OrderCard key={order.id} {...order} />)
        ) : (
          <p>ไม่มีข้อมูล</p>
        )}
      </div>
    </div>
  );
}

```

ภาพที่ 4.1.4 แสดงหน้าดูประวัติการซื้อและตัวอย่าง Code

จากภาพที่ 4.1.4 แสดงหน้าประวัติการซื้อที่จะมีรายการสินค้าที่เราเคยซื้อหรือกำลังอยู่ในกระบวนการซื้อ ซึ่งเราระบุรายการสินค้าตามสถานะของสินค้านั้น

Input: ผู้ใช้คลิกดูประวัติคำสั่งซื้อ และสามารถเลือกดูรายการสินค้าตามสถานะที่เลือก  
Process: ระบบดึงข้อมูลคำสั่งซื้อของผู้ใช้นั้นตามสถานะที่เลือก

Output: แสดงรายการสั่งซื้อพร้อมรายละเอียดสินค้า, สถานะ และวันที่สั่งซื้อ

#### 4.1.5 UC05: Add to Basket

The screenshot displays the Farmly Neighbor mobile application interface. At the top, there's a search bar and navigation buttons. The main content area shows a product detail page for 'เมล่อน' (Melon) with a price of \$5555.00. A quantity selector shows '2'. Below the price, there's a green button labeled 'เพิ่มลงตะกร้า' (Add to basket). The background shows a shopping cart icon with the number '1&1'.

```
<DrawerContent className="w-80 max-w-full">
  <DrawerHeader>
    <DrawerTitle>เพิ่มลงตะกร้า</DrawerTitle>
    <DrawerDescription>รายการสินค้าที่ต้องการ</DrawerDescription>
  </DrawerHeader>

  <div className="p-4 space-y-4 max-h-[60vh] overflow-auto">
    {items.length === 0 ? (
      <p className="text-sm text-muted-foreground">ไม่มีสินค้าในตะกร้า</p>
    ) : (
      items.map((item, i) => (
        <Card key={i} className="flex items-center p-3">
          <CardContent className="p-0 flex flex-col gap-1 w-full">
            <div className="flex justify-between">
              <img
                src={item.image || "https://upload.wikimedia.org/wikipedia/commons/c/cd/1%261_logo.svg"}
                alt={item.name}
                className="w-24 h-24 object-cover rounded-md"
              />
              <div className="flex flex-col flex-1 px-4">
                <p className="font-medium w-full">{item.name}</p>
                <p className="text-muted-foreground text-xs">×{item.quantity}{item.unit} - ${({item.price * item.quantity}).toFixed(2)}</p>
              </div>
              <Button
                variant="ghost"
                size="icon"
                aria-label="Remove item"
                onClick={() => removeItem(i)}
              >
                <Trash2 className="w-4 h-4" />
              </Button>
            </div>
          </CardContent>
        </Card>
      )));
  </div>
</DrawerContent>
```

ภาพที่ 4.1.5 แสดงหน้าสินค้าที่สามารถกดเพิ่มสินค้าลงตะกร้าและตัวอย่าง Code

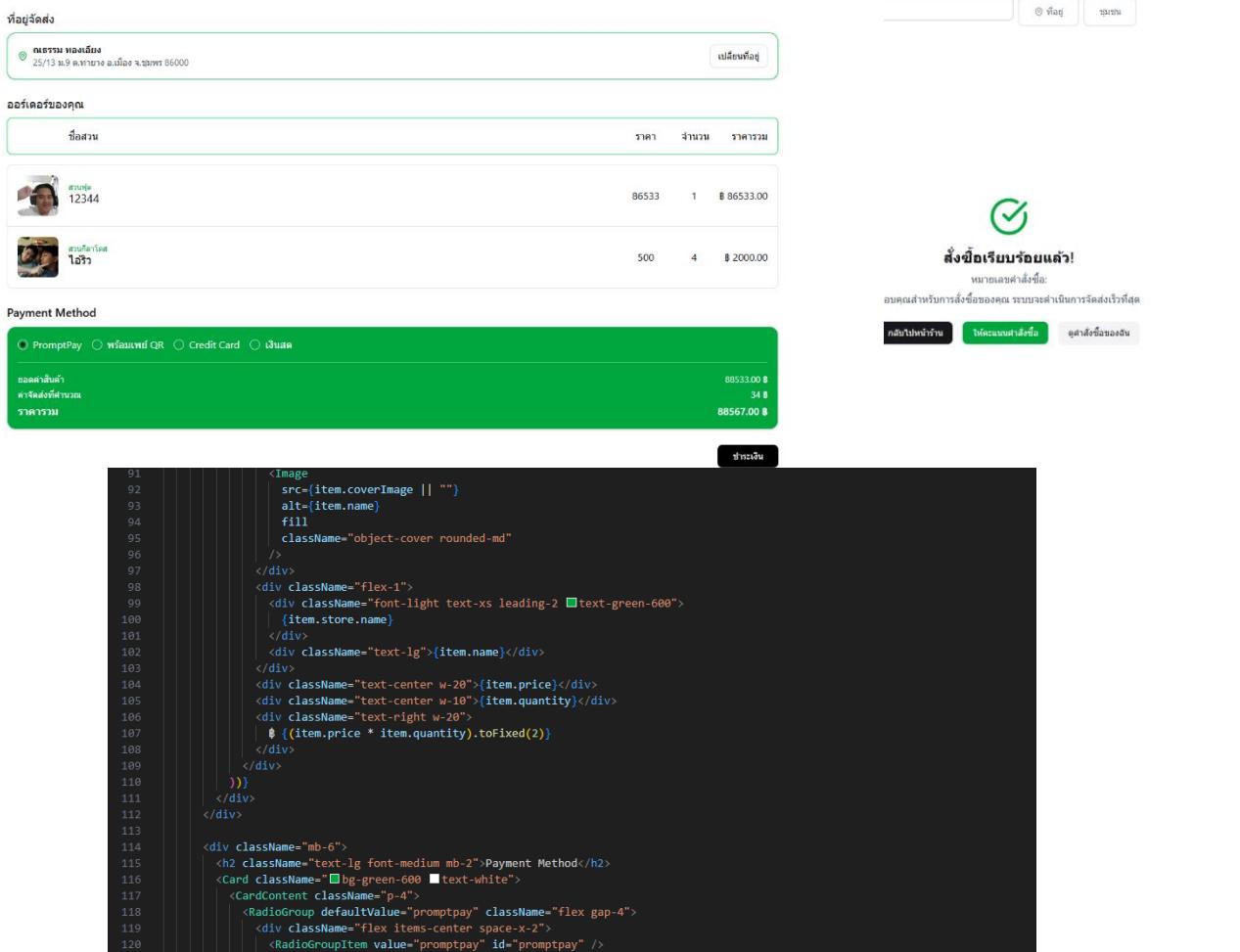
จากภาพที่ 4.1.5 แสดงหน้าสินค้าที่มีรายละเอียดสินค้าต่างๆ แล้วก็สามารถเลือกจำนวนสินค้าที่เราต้องการและกดเพิ่มลงตะกร้า แล้วสินค้าจะเข้าสู่หน้าตะกร้าแล้วจะสามารถกดชำระเงินได้

Input: ผู้ใช้เลือกสินค้าที่ต้องการจากหน้าแรก แล้วเลือกจำนวนที่ต้องการซื้อ

Process: เมื่อคลิก “เพิ่มลงตะกร้า” ระบบจะบันทึกรายการสินค้านั้นลงในตะกร้าของผู้ใช้

Output: ระบบเพิ่มสินค้าเข้าในตะกร้า และสามารถไปหน้าตะกร้าเพื่อชำระเงินต่อได้

#### 4.1.6 UC06: Check Out Product in Basket



The screenshot shows a mobile application interface for checking out products from a basket. At the top, there is a header with a profile picture, name (นายสมชาย ใจดี), and address (25/13 ม.9 ถ.พานาธิ ถนนสุขุมวิท กรุงเทพฯ 10110). Below the header is a search bar and a button labeled "เข้าสู่ระบบ".

The main content area displays a shopping basket table:

รายการ	ราคา	จำนวน	รวม
สมชาย ใจดี 12344	86533	1	฿ 86533.00
สมชาย ใจดี ใจดี	500	4	฿ 2000.00
<b>รวมทั้งหมด</b>	<b>86533.00</b>	<b>34</b>	<b>฿ 88567.00</b>

Below the basket, there is a section for "Payment Method" with options: PromptPay (selected), WeChat QR, Credit Card, and ATM. A green button labeled "ชำระเงิน" (Pay) is visible.

On the right side, there is a success message: "สั่งซื้อเรียบร้อยแล้ว!" (Order placed successfully!) with a checkmark icon. It also includes a note: "หากมีปัญหาติดต่อเราได้ที่: อีเมลล์: support@sampleapp.com หรือโทรศัพท์: 098-123-4567".

At the bottom, there is a code editor window showing the source code for the payment method selection screen:

```

91     <Image
92       src={item.coverImage || ""}
93       alt={item.name}
94       fill
95       className="object-cover rounded-md"
96     />
97   </div>
98   <div className="flex-1">
99     <div className="font-light text-xs leading-2 text-green-600">
100       {item.store.name}
101     </div>
102     <div className="text-lg">{item.name}</div>
103     <div className="text-center w-20">{item.price}</div>
104     <div className="text-center w-10">{item.quantity}</div>
105     <div className="text-right w-20">
106       ${((item.price * item.quantity).toFixed(2))}
107     </div>
108   </div>
109   <div>
110     <div>
111       <div><h2>Payment Method</h2><Card className="bg-green-600 text-white"><CardContent className="p-4">
112         <RadioGroup defaultValue="promptpay" className="flex gap-4">
113           <div className="flex items-center space-x-2">
114             <RadioGroupItem value="promptpay" id="promptpay" />
115           </div>
116         </RadioGroup>
117       </div>
118     </div>
119   </div>
120 
```

ภาพที่ 4.1.6 แสดงหน้าคิดตั้งและตัวอย่าง Code

จากภาพที่ 4.1.6 แสดงหน้าคิดตั้งโดยจะเป็นรายการสินค้าที่อยู่ในตะกร้า แล้วเมื่อเราเลือกวิธีการชำระแล้วกดชำระเงิน ระบบก็บันทึกการทำรายการและแจ้งเตือนว่าสั่งซื้อเรียบร้อย

Input: ผู้ใช้เลือกสินค้าที่อยู่ในตะกร้า และเลือกวิธีการชำระเงิน ( เช่น โอนเงิน )

Process: ระบบตรวจสอบตะกร้า และเมื่อยืนยันการสั่งซื้อ ระบบจะบันทึกคำสั่งซื้อไว้ในฐานข้อมูล

Output: ระบบแสดงข้อความ “สั่งซื้อสำเร็จ” และอัปเดตสถานะของเดอร์ในระบบ

#### 4.1.7 UC07: Search Item

```

    ...
    include: {
      images: true,
      category: true,
      store: {
        select: [
          { name: true },
        ],
      },
    },
    orderBy: {
      createdAt: "desc",
    },
  });

  if (products.length === 0) return <p className="p-4">ໃນເພີ້ມເລີນຄ້າ</p>;
}

return (
  <div className="mx-auto max-w-7xl">
    <Navbar />
    <MiniNav />
    <BannerSection />
    <div className="grid grid-cols-[.15fr_.9fr]">
      <div className="w-full pl-4">
        <MinIMenus categories={categories} />
      </div>
      <div className="grid grid-cols-4 mx-auto gap-6 mt-4">
        <ProductSection products={products} />
      </div>
      <BenefitsSection />
    </div>
  </div>
)

```

ກາພທີ 4.1.7 ແສດໜ້າຫຼັກທີ່ສາມາຮັດຄົນຫາຫຼືຈັດເຮືອງສິນຄ້າໄດ້ແລະຕ້ວຍ່າງ Code

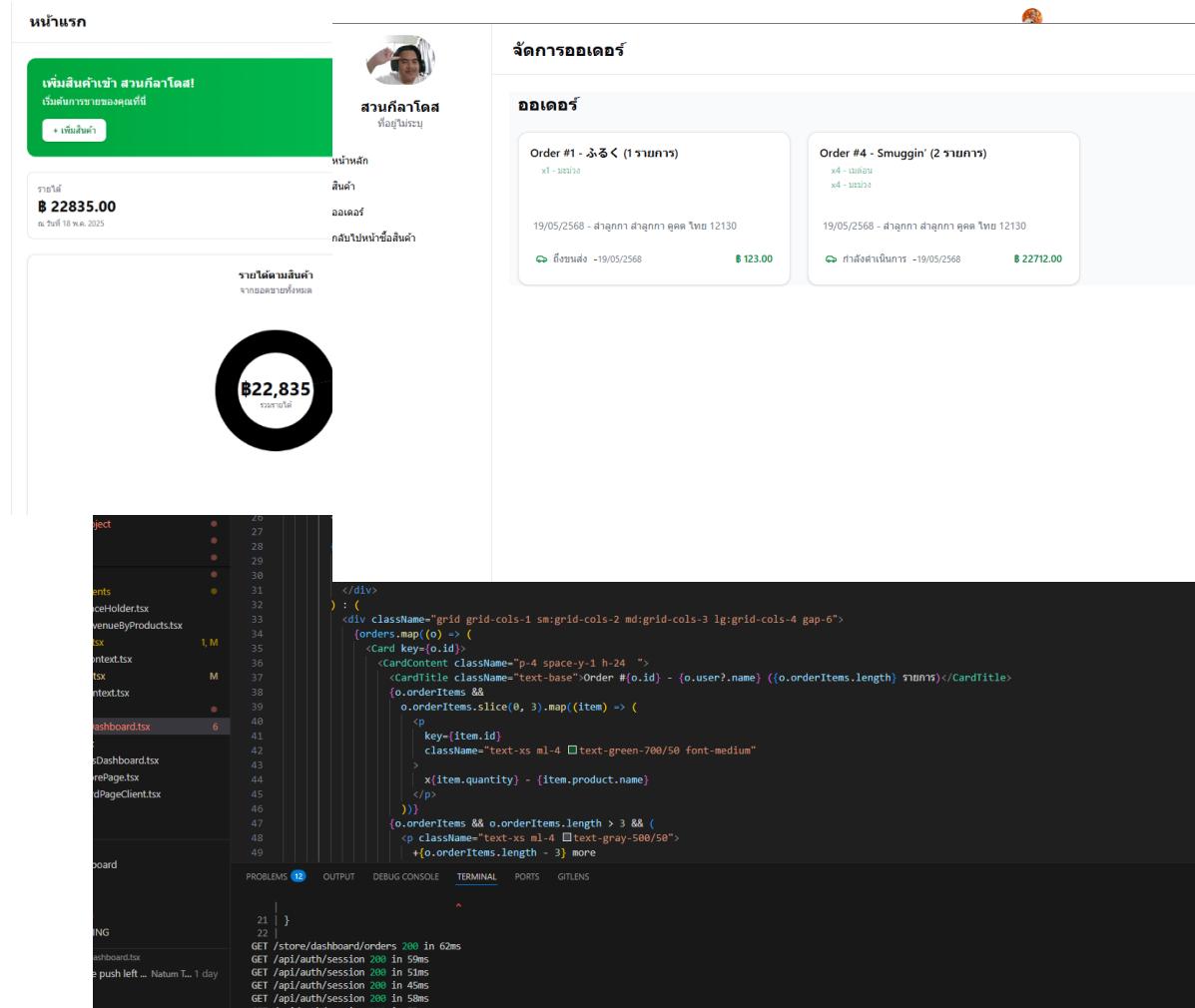
ຈາກກາພທີ 4.1.7 ໃນໜ້າຫຼັກເຮົາສາມາຮັດກົດແທບທາງຂວາມື່ອເພື່ອຈັດເຮືອງສິນຄ້າຕາມແຕ່ລະປະເກດແລະ ຍັງມີແບບຄົນຫາທີ່ສາມາຮັດໄສ່ຄໍາເພື່ອຄົນຫາສິນຄ້າຕາມຄໍາທີ່ໄສ່ລັງໄປ

Input: ຜູ້ໃຊ້ສາມາຮັດເລືອກໝວດໝູ່ທາງດ້ານຂວາ ຫຼືພິມພົບຄຳຄັນໃນໜ້າຄົນຫາ

Process: ຮະບບກຮອງຫຼືຄົນຫາສິນຄ້າຕາມຄໍາທີ່ປຸ່ນ

Output: ແສດຮາຍກາຮັດສິນຄ້າທີ່ໂຕຮັດກັບໝວດໝູ່ຫຼືຄຳຄັນທີ່ກຮອກໄວ້

#### 4.1.8 UC08: Check Farmer History



ภาพที่ 4.1.8 แสดงหน้าดูอุตสาหกรรมสินค้าของเกษตรกรและตัวอย่าง Code

จากภาพที่ 4.1.8 แสดงหน้าดูอุตสาหกรรมสินค้าของเกษตรกร เมื่อเข้ามาหน้าจัดการร้านค้าแล้วเกษตรจะสามารถเห็นถึงรายได้และจำนวนออเดอร์ เมื่อกดเข้าไปที่ ออเดอร์ จะสามารถดูรายละเอียดต่างๆของ ออเดอร์ได้

**Input:** เกษตรกรเข้าสู่ระบบและกดเข้าไปที่ “ร้านค้า” จากนั้นเลือกดูรายได้หรือรายการออเดอร์

**Process:** ระบบดึงข้อมูลรายได้และออเดอร์ทั้งหมดของร้านนั้นจากฐานข้อมูล

**Output:** แสดงรายได้รวม และรายการออเดอร์แต่ละรายการ (รวมถึงรายละเอียดออเดอร์ เช่น ชื่อผู้ สั่ง, จำนวน, สถานะ)

#### 4.1.9 UC09: Add Product (Agriculture)

The screenshot shows the Family Neighbor application interface. On the left, there's a sidebar with options like 'Store' and 'Avatar'. The main area has two windows. One window is titled 'เพิ่มสินค้า ร้านพ่อ' (Add Product) with a sub-instruction 'เริ่มเพิ่มรายการของผลผลิต'. The other window shows a list of products with columns for 'ชื่อสินค้า', 'ราคาส่งขาย', 'หน่วยนับ', 'ประเภท', 'หมวดหมู่', and 'จำนวน'. Below these windows is a code editor showing the 'AddProduct.tsx' component code.

```

    export default function AddProduct() {
      const handleSubmit = async (e: React.FormEvent) => {
        try {
          const res = await fetch("/api/addproduct", {
            method: "POST",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify({
              productName: formData.productName,
              price: Number(formData.price),
              unit: formData.unit,
              type: parseInt(formData.type), // แปลงเป็น number
              stock: Number(formData.stock),
              description: formData.description,
              storeId: Number(storeId),
            }),
          });
          const data = await res.json();
          if (data.ok) {
            setApiResponse(data.message || "เพิ่มสินค้า thành công");
          } else {
            setApiResponse("เพิ่มสินค้าไม่สำเร็จ");
            setFormData({
              productName: "",
              price: 0,
              unit: "",
              type: "",
              stock: 0,
              description: "",
            });
            setTimeout(() => {
              router.push(`/${storeId}`);
            }, 2000);
          }
        } catch (err) {
          console.error(err);
        }
      };
    }
  
```

ภาพที่ 4.1.9 แสดงหน้าร้านค้าที่สามารถเพิ่มสินค้าได้และตัวอย่าง Code

จากภาพที่ 4.1.9 แสดงหน้าร้านค้าของเกษตรกรที่จะมีรายการสินค้าที่เกษตรกรลงขายไว้แล้วรายจังสามารถกดเพิ่มสินค้า เมื่อเรากรอกข้อมูลครบก็จะสร้างรายการสินค้าขึ้นมา

Input: เกษตรกรคลิก “เพิ่มสินค้า” และกรอกข้อมูลสินค้า (ชื่อ, ราคา, รูปภาพ, คำอธิบาย ฯลฯ)

Process: ระบบตรวจสอบความครบถ้วนของฟอร์มและบันทึกข้อมูลลงฐานข้อมูล

Output: สินค้าใหม่จะถูกเพิ่มเข้ารายการสินค้าในร้านค้าของเกษตรกร และแสดงในหน้าสินค้าสาขาแรก

#### 4.1.10 UC10: Community Farmer

```

import { prisma } from "@/lib/prisma";
import { notFound } from "next/navigation"; 34.2k (gzipped: 7.5k)
import CommentForm from "./CommentForm";

export default async function ThreadDetailPage({ params }: { params: { id: string } }) {
  const threadId = Number(params.id);
  if (isNaN(threadId)) return notFound();

  const thread = await prisma.thread.findUnique({
    where: { id: threadId },
    include: [
      { author: true },
      { comments: {
          include: { author: true },
          orderBy: { createdAt: "asc" },
        }, take: 2, dayAgo: true, count: true
      },
    ],
  });
  if (!thread) return notFound();

  return (
    <div className="max-w-3xl mx-auto px-4 py-8">
      <h1 className="text-2xl font-bold text-green-700 mb-2">{thread.title}</h1>
      <p className="text-sm text-gray-500 mb-4">by {thread.author.name} · (new Date(thread.createdAt).toLocaleString())</p>
      <p className="mb-6">{thread.content}</p>
      <h2 className="text-lg font-semibold mb-2">ความคิดเห็น ({thread.comments.length})</h2>
      <div className="space-y-4 mb-6">
        {thread.comments.map((comment) => (
          <div key={comment.id} className="border p-3 rounded-3g">
            <p className="text-sm text-gray-800">comment.content</p>
            <p className="text-xs text-gray-500">by {comment.author.name} · (new Date(comment.createdAt).toLocaleString())</p>
          </div>
        ))
      </div>
    </div>
  );
}

```

ภาพที่ 4.1.10 แสดงหน้าชุมชนและตัวอย่าง Code

จากภาพที่ 4.1.10 แสดงหน้าชุมชนที่สามารถตั้งกระทู้เพื่อมาพูดคุย โดยจะมีรายการกระทู้และสามารถเข้ากระทู้เพื่อเข้าไปแสดงความคิดเห็นได้

Input: ผู้ใช้เข้าสู่หน้าชุมชน และสามารถตั้งกระทู้ใหม่โดยพิมพ์หัวข้อและเนื้อหา หรือเลือกกระทู้เก่า เพื่อแสดงความคิดเห็น

Process: สำหรับการตั้งกระทู้: ระบบบันทึกชื่อกระทู้และเนื้อหาใหม่

สำหรับการแสดงความคิดเห็น: ระบบบันทึกความคิดเห็นไว้ใต้กระทู้นั้น

Output: กระทู้ใหม่จะแสดงในรายการกระทู้ ความคิดเห็นใหม่จะแสดงใต้โพสต์ต้นเรื่องของกระทู้

## 4.2 การทดสอบระบบเบื้องต้น

หลังจากดำเนินการพัฒนาเว็บไซต์ซื้อขายสินค้าเกษตรในระดับที่สามารถใช้งานได้แล้ว ได้มีการดำเนินการทดสอบระบบเบื้องต้นโดยอาศัยการทดสอบแบบ User-based Testing โดยเชิญผู้ที่มีทักษะทางด้านการพัฒนาเว็บไซต์ (Developer) มาช่วยทดลองใช้งานและให้คำแนะนำในเบื้องต้น เพื่อประเมินความถูกต้องของระบบ การทำงานของฟังก์ชัน และประสบการณ์ใช้งาน (UX/UI) ของผู้ใช้

### -วิธีการทดสอบ

- ผู้พัฒนาทำการเตรียมระบบในสภาพแวดล้อมการใช้งานจริง (Development Build)

- ให้ผู้ทดสอบ (เป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาการคอมพิวเตอร์ที่มีประสบการณ์ในการพัฒนาระบบออนไลน์) ลองใช้งานระบบในฐานะ “ผู้ใช้ทั่วไป”

- ให้ทดลองใช้ฟังก์ชันต่าง ๆ เช่น การสมัครสมาชิก การเข้าสู่ระบบ การเลือกซื้อสินค้า การเพิ่มสินค้าในตะกร้า การดูประวัติคำสั่งซื้อ และการเขียนรีวิว

- สังเกตพฤติกรรมการใช้งาน และสอบถามความคิดเห็นในเบื้องต้นหลังการทดลอง

การทำงาน	วิธีการทดสอบ	ข้อมูลนำเข้า	ผลที่คาดหวัง	ผลที่ได้รับ	ข้อจำกัด
การสมัครสมาชิก	ให้ผู้ทดสอบลงทะเบียนผ่านหน้า Register	อีเมล, รหัสผ่าน, เบอร์โทรศัพท์	สมัครสำเร็จ ระบบสร้างบัญชีผู้ใช้ใหม่	สมัครสำเร็จ ไม่มี Error	ยังไม่มีระบบยืนยันว่าเป็นข้อมูลของจริง รีปล่า
การเข้าสู่ระบบ	ผู้ทดสอบเข้าสู่ระบบด้วยบัญชีที่ลงทะเบียนไว้	อีเมล + รหัสผ่าน	เข้าสู่ระบบได้ เข้าถึงข้อมูลผู้ใช้	ระบบเข้าสู่ระบบได้ตามปกติ	ยังไม่มีระบบยืนยันตัวตน 2 ชั้น (2FA)

เลือกซื้อสินค้า	ผู้ทดสอบ เลือกดูสินค้า จากหน้า Homepage	เลือกสินค้าใน แต่ละ หมวดหมู่	สินค้าแสดง ถูกต้อง สามารถคลิก ดูรายละเอียด	การแสดงผล สินค้า เรียบร้อย	ไม่ได้ทดสอบ กับสินค้า จำนวนมาก ยังไม่ทราบว่า UI จะช้าไหม
เพิ่มสินค้าใน ตะกร้า	ทดลองเพิ่ม สินค้าโดย ปรับจำนวน ก่อนกด "เพิ่ม ลงตะกร้า"	จำนวน สินค้า 1-5 ชิ้น	สินค้าเพิ่มใน ตะกร้า พร้อม คำนวนราคา ถูกต้อง	เพิ่มใน ตะกร้าได้ ระบบ คำนวนราคา ถูกต้อง	เวลาช้าระเงิน เสร็จแล้วไม่ ลบสินค้าออก จากตะกร้า
ดูประวัติคำ สั่งซื้อ	ทดลองสั่งซื้อ และเข้าหน้า History เพื่อ ตรวจสอบ	คำสั่งซื้อ จำนวน 1-2 รายการ	แสดงคำสั่งซื้อ <sup>*</sup> ย้อนหลัง พร้อม รายละเอียด	ระบบแสดง รายการได้ ถูกต้อง	
เขียนรีวิว	ทดสอบการ แสดงความ คิดเห็นหลัง สั่งซื้อ	ข้อความรีวิว, rating	รีวิวถูก <sup>*</sup> บันทึกและ แสดงในหน้า product	รีวิวแสดง ผลได้จริง	ยังไม่มีการ จำกัดความ ยาวของ ข้อความและ ยังไม่มีพิม รูป
สั่งซื้อบทบาท ผู้ใช้	ผู้ทดสอบ เปลี่ยน สถานะจาก Customer → Seller	กดปุ่ม <sup>*</sup> ไอคอน <sup>*</sup> “ร้านค้า” Navbar	ระบบเปลี่ยน บทบาทและ แสดงหน้า <sup>*</sup> “จัดการ สินค้า”	เปลี่ยน บทบาทได้ จริง พิงก์ชัน ขายสินค้าใช้ งานได้	ยังไม่ได้ ทดสอบการ จัดการ order ในฐานะ seller อย่าง ละเอียด

การสร้าง กราฟท์และ เขียนความ คิดเห็นใน ชุมชน	ผู้ทดสอบกด สร้างกราฟท์ ใหม่และเขียน ความคิดเห็น	ข้อความ กราฟท์และ ความคิดเห็น	มีกราฟท์ใน หน้าชุมชนและมี ความคิดเห็น ในกราฟท์	กราฟท์ในหน้า ชุมชนและมี ความคิดเห็น ในกราฟท์ ตามที่หวัง	ยังไม่มีเพิ่มรูป <sup>ในกราฟท์และ ความคิดเห็น</sup>

ตารางที่ 4.2 ผลการทดสอบระบบเบื้องต้น

-ผลการทดสอบ

- ระบบสามารถใช้งานได้จริงโดยไม่พบ Error ที่ร้ายแรง
- พังก์ชันหลักทำงานได้ครบถ้วนตามที่ตั้งใจไว้ เช่น การเพิ่มสินค้า การสั่งซื้อบาท ผู้ใช้ และการบันทึกคำสั่งซื้อ
  - UI มีความเรียบง่ายและเข้าใจง่ายในภาพรวม
  - ผู้ทดสอบเสนอให้ปรับปรุงในบางส่วน เช่น ยังมีบางจุดที่ยัง Bung อยู่ เช่น UI ซ้อนกัน บ้างหรือการที่ประสิทธิภาพของเว็บนั้นไม่ได้ดีขนาดนั้น หรือจะเป็นเรื่องความสะดวกสบายที่สามารถทำได้ดีกว่านี้
    - พบร่วงการบางพังก์ชันอาจยังทำให้ผู้ใช้สับสนเล็กน้อย จึงควรพิจารณาออกแบบ UX/UI ให้ชัดเจนยิ่งขึ้นในอนาคต

-ข้อจำกัดของการทดสอบ

เนื่องจากเวลาจำกัด จึงยังไม่มีโอกาสเก็บข้อมูลการทดสอบจากผู้ใช้งานทั่วไป (Non-developer) หรือกลุ่มเป้าหมายที่เป็นเกษตรกรจริง ทำให้การประเมินด้านความเข้าใจของผู้ใช้ทั่วไป ยังไม่ครบถ้วน ซึ่งหากมีโอกาสพัฒนาต่อในอนาคต ควรดำเนินการทดสอบอย่างเป็นระบบมากยิ่งขึ้น เช่น:

- การทดสอบแบบ Usability Testing กับกลุ่มเป้าหมายจริง
- การเก็บแบบสอบถามความพึงพอใจ
- การใช้เครื่องมือวัดพฤติกรรมการใช้งาน (เช่น heatmap, analytics)

## บทที่ 5

### สรุป

จากการดำเนินการพัฒนาเว็บไซต์ซื้อขายสินค้าเกษตรออนไลน์ตลอดระยะเวลาโครงการ พบร่วมกันสามารถพัฒนาระบบได้ครบถ้วนตามวัตถุประสงค์หลักที่ตั้งไว้ โดยสามารถสรุปผลการพัฒนาได้ในประเด็นสำคัญ ดังนี้

#### 5.1 ความสมบูรณ์ของฟังก์ชัน

ระบบที่พัฒนาแล้วสามารถใช้งานได้จริงและครอบคลุมฟังก์ชันหลักที่จำเป็นสำหรับแพลตฟอร์ม e-commerce ได้แก่:

- ระบบ Log in – Log out : ผู้ใช้สามารถเข้าสู่ระบบและออกจากระบบได้
- การ Switch Role : ผู้ใช้ที่เป็นเกษตรลสัปบไปมาระหว่างบทบาทได้
- ระบบแสดงรายการสินค้า: ผู้ใช้งานสามารถเรียกดูสินค้าแบบ Grid พร้อมกรองตามหมวดหมู่ได้
- หน้ารายละเอียดสินค้า: แสดงข้อมูลสินค้าอย่างครบถ้วน เช่น รูปภาพ ราคา คำอธิบาย หน่วยสินค้า และข้อมูลร้านค้า
- ระบบตั้งกร้าสินค้าและคำสั่งซื้อ: ผู้ใช้สามารถเพิ่มสินค้าลงตั้งกร้า กำหนดจำนวน และดำเนินการสั่งซื้อ โดยมีการบันทึกคำสั่งซื้อในฐานข้อมูล
- ระบบรีวิวสินค้า: ผู้ใช้สามารถแสดงความคิดเห็นและให้คะแนนสินค้าที่ซื้อไปแล้ว
- ระบบชุมชน (Community): ผู้ใช้สามารถตั้งกระทู้ถาม-ตอบ และแสดงความคิดเห็น หรือแบ่งปันข้อมูลในรูปแบบกระทู้ได้
- ระบบໂປຣີເພີ້ໃຊ້: ผู้ใช้สามารถดูและแก้ไขข้อมูลส่วนตัว
- ระบบ Add Product (Agriculture) : เกษตรสามารถเพิ่มสินค้าลงขายได้
- ระบบ Check Buy History : ผู้ใช้สามารถดูประวัติการซื้อได้

## 5.2 เทคโนโลยีและเครื่องมือที่ใช้

การพัฒนาใช้เทคโนโลยีที่ทันสมัยและได้รับความนิยม ทำให้ระบบสามารถปรับขยายได้ง่ายและมีความปลอดภัย:

- Frontend: ใช้ Next.js (App Router) ในการสร้าง UI แบบ Component-based ที่ทำงานได้รวดเร็วและยืดหยุ่น
- Backend: ใช้ API Route ของ Next.js ในการรับส่งข้อมูลผ่าน REST API พร้อมระบบ Authentication
- Database: ใช้ PostgreSQL ในการจัดเก็บข้อมูล และ Prisma ORM เพื่อจัดการความซ้ำซ้อนของข้อมูลอย่างมีประสิทธิภาพ
- Authentication: ใช้ NextAuth.js รองรับ OAuth (Google, Facebook, X) และ Credential (อีเมล/รหัสผ่าน) พร้อมระบบ JWT สำหรับการจัดการ session

## 5.3 ความสอดคล้องกับเป้าหมายโครงการ

ระบบที่พัฒนาขึ้นสามารถตอบโจทย์ตามเป้าหมายของโครงการ ได้แก่

- สร้างช่องทางออนไลน์ในการซื้อขายสินค้าเกษตร
- เพิ่มโอกาสให้เกษตรกรมีพื้นที่จำหน่ายสินค้าด้วยตนเอง
- อำนวยความสะดวกให้ผู้บริโภคสามารถเข้าถึงสินค้าเกษตรได้ง่ายขึ้น
- สร้างระบบที่สามารถต่อยอดพัฒนาได้ในอนาคต เช่น เพิ่มระบบขนส่ง หรือระบบวิเคราะห์ข้อมูล

## รายการอ้างอิง

- Asian Digital Hub. (2024, February 15). *What is Next.js and its benefits?*. Medium. <https://medium.com/devsphere/what-is-next-js-and-its-benefits-8b13aab56bfd>
- Whitfield, B. (2025, April 18). *What Is HTML?*. Built In. <https://builtin.com/software-engineering-perspectives/html>
- Lutkevich, B. (2024, Jul 09). *What is GitHub?*. TechTarget. <https://www.techtarget.com/searchitoperations/definition/GitHub>
- Voraprateep, K. (2023, November 28). เบรี่บเที่ยบการใช้งาน CSS แบบต่างๆ. BorntoDev. <https://www.borntodev.com/2023/11/28/เบรี่บเที่ยบการใช้งาน-css/>
- ปานนัท, ย. (2021, July 13). ผลสำรวจพฤติกรรมเกษตรกรต่อการใช้เทคโนโลยี. Maejo Poll. <https://maejopoll.mju.ac.th/poll.aspx?id=2157>
- ปานนัท, ย. (2022, November 15). ความคิดเห็นของเกษตรกรต่อการขายสินค้าออนไลน์. Maejo Poll. <https://maejopoll.mju.ac.th/poll.aspx?id=4241>
- ชนมานะวัตร, ม. (2019). พฤติกรรมการใช้อินเทอร์เน็ตเพื่อการเกษตรของเกษตรกรไทย [วิทยานิพนธ์ปริญญาโท, มหาวิทยาลัยมหิดล]. Mahidol Institutional Repository. <https://archive.cm.mahidol.ac.th/bitstream/123456789/2969/1/TP%20FB.009%202562.pdf>
- Mikelopster. (2024, February 22). รู้จักกับ Prisma ORM. Mikelopster Blog. <https://mikelopster.dev/posts/next-prisma/>
- Mikelopster. (2024, January 11). ลองเล่น Supabase กับ Next.js กัน. Mikelopster Blog. <https://mikelopster.dev/posts/supabase-next/1>
- Mitr Phol Modern Farm. (2022, May 21). มาตรฐานสินค้าเกษตรสำคัญอย่างไร และมีอะไรบ้าง. <https://www.mitrapholmodernfarm.com/news/2022/05/มาตรฐานสินค้าเกษตรสำคัญอย่างไร-และมีอะไรบ้าง>

PostgreSQL. (n.d.). *What is PostgreSQL?*.

<https://www.postgresql.org/about/>

Rakbankerd. (2020, October 7). 10 Platform Online ช่วยกรະຈາຍລືບນົກ້າເກຫດຮ່າງໄທຍ. <https://www.rakbankerd.com/agriculture/hilight-view.php?id=238&s=tblheight>

Sueakkam, S. (2024, April 5). การໃຊ້ງານ CSS ໃນການປະຕິບັດແຕ່ງໜ້າເວັບ. BorntoDev. <https://www.borntodev.com/2024/04/05/การໃຊ້ງານ-css/>

ສາຂາວິທາຄາສດຖະກິນ ມາຮະວິທາລີຍເກຫດຮ່າງໄທຍ. (2021). ແນວໃນ້ມະແລະພັດທິກຣມຂອງເກຫດຮ່າງໄທຍໃນຍຸດທິຈິຫຼ້ລ. Agricultural Science Journal, 52(1). <http://agscij.agr.ku.ac.th/phocadownload/2564-52-1/ASJ-52-1-2.pdf>