

การพัฒนาแพลตฟอร์มสำหรับการซื้อขายผ่านช่องทางออนไลน์

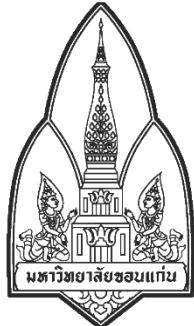
Development of an E-commerce Platform

โดย

663380395-4	นางสาวพิชามณฑ์	พงศ์ศรีษฐ์สันต์	กลุ่มการเรียนที่ 4
663380399-6	นายวราชาติ	ปรัญญา	กลุ่มการเรียนที่ 3
663380404-9	นายศุภกฤต	แก้วแภรณ์ทอง	กลุ่มการเรียนที่ 3
663380599-8	นายธิติกร	สุวรรณบุตรวิภา	กลุ่มการเรียนที่ 4
663380611-4	นายภูชิต	จำปาชน	กลุ่มการเรียนที่ 3

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา CP353002 หลักการพัฒนาซอฟต์แวร์
ภาคเรียนที่ 1 ปีการศึกษา 2568

สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์
มหาวิทยาลัยขอนแก่น



การพัฒนาแพลตฟอร์มสำหรับการซื้อขายผ่านช่องทางออนไลน์

Development of an E-commerce Platform

โดย

663380395-4	นางสาวพิชามณฑ์ พงศ์เศรษฐสันต์	กลุ่มการเรียนที่ 4
663380399-6	นายวรชาติ ปรัญญา	กลุ่มการเรียนที่ 3
663380404-9	นายศุภกฤต แก้วแคมทอง	กลุ่มการเรียนที่ 3
663380599-8	นายธิติกร สุวรรณบุตรวิภา	กลุ่มการเรียนที่ 4
663380611-4	นายภูชิต จำปาชน	กลุ่มการเรียนที่ 3

เสนอ

รศ. ดร.ปัญญาพล หอระตะ

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา CP353002 หลักการพัฒนาซอฟต์แวร์

ภาคเรียนที่ 1 ปีการศึกษา 2568

สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์

มหาวิทยาลัยขอนแก่น

ชื่อหัวข้อโครงการ	การพัฒนาแพลตฟอร์มสำหรับการซื้อขายผ่านช่องทางออนไลน์		
ชื่อผู้จัดทำ	นางสาวพิชามณฑ์ พงศ์เศรษฐสันต์	บัตรประชาชน	663380395-4
	นายวรชาติ ประญญา	บัตรประชาชน	663380399-6
	นายศุภกฤต แก้วแคมทอง	บัตรประชาชน	663380404-9
	นายธิติกร สุวรรณบุตรวิภา	บัตรประชาชน	663380599-8
	นายภูษิต จำปาชน	บัตรประชาชน	663380611-4
อาจารย์ที่ปรึกษา	รศ. ดร.ปัญญา พล หอระตะ		
สาขาวิชา	วิทยาการคอมพิวเตอร์ วิทยาลัยการคุณพิวเตอร์		
ภาคการศึกษา	1/2568		

บทคัดย่อ

โครงการนี้มีวัตถุประสงค์เพื่อพัฒนาระบบเว็บไซต์พาณิชย์อิเล็กทรอนิกส์ (E-commerce) ที่รองรับผู้ใช้งานใน 3 บทบาทหลัก ได้แก่ ผู้ซื้อ (Buyer) ผู้ขาย (Seller) และผู้ดูแลระบบ (Admin) โดยมุ่งเน้นการสร้างแพลตฟอร์มที่มีประสิทธิภาพและมีความปลอดภัยสูง พร้อมทั้งระบบการสื่อสารระหว่างผู้ซื้อและผู้ขายในรูปแบบเรียลไทม์

ในการดำเนินการพัฒนาแบ่งออกเป็น 2 ส่วนหลัก ได้แก่ ส่วน Frontend ซึ่งใช้ Next.js ร่วมกับ Tailwind CSS เพื่อสร้างประสบการณ์การใช้งานที่ทันสมัยและตอบสนองได้รวดเร็ว ส่วน Backend พัฒนาด้วย Spring Boot โดยใช้ PostgreSQL สำหรับการจัดการฐานข้อมูล, JPA สำหรับการจัดการ Object-Relational Mapping, JWT สำหรับการยืนยันตัวตนและการรักษาความปลอดภัย, CORS สำหรับการจัดการคำขอข้ามโดเมน และระบบ Logging สำหรับการติดตามการทำงานของระบบ

จากการพัฒนา ระบบสามารถรองรับการทำงานของผู้ใช้งานทั้ง 3 บทบาทได้อย่างสมบูรณ์ ผู้ซื้อสามารถเลือกซื้อสินค้า เขียนรีวิว ชำระเงิน และติดตามสถานะการจัดส่ง ผู้ขายสามารถเปิดร้านค้า ตรวจสอบคำสั่งซื้อ เพิ่มสินค้า และจัดการรีวิวจากลูกค้าได้ และผู้ดูแลระบบมีสิทธิ์ในการแบบร้านค้าที่ฝ่ายนักกฎหมาย นอกจากนี้ยังมีฟีเจอร์การแข่งขันแบบเรียลไทม์ระหว่างผู้ซื้อและผู้ขาย เพื่ออำนวยความสะดวกในการสื่อสาร

แม้ว่าระบบจะมีความเสถียรและปลอดภัย สามารถรองรับการทำงานพื้นฐานของแพลตฟอร์มได้ครบถ้วน แต่ยังมีข้อจำกัดในส่วนของระบบแจ้งเตือนที่ยังไม่ได้รับการพัฒนา โดยในอนาคตควรมีการเพิ่มระบบแจ้งเตือนเพื่อเพิ่มการมีส่วนร่วมของผู้ใช้ พร้อมทั้งพัฒนาฟีเจอร์เพิ่มเติม เช่น ระบบแนะนำสินค้า การวิเคราะห์พฤติกรรมผู้ใช้ และระบบรายงานสำหรับผู้ขาย เพื่อให้แพลตฟอร์มสามารถตอบโจทย์และขับเคลื่อนธุรกิจได้อย่างมีประสิทธิภาพยิ่งขึ้น

กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จลุล่วงได้ด้วยความร่วมมือและการสนับสนุนจากหลายฝ่าย ผู้จัดทำขอขอบพระคุณ อาจารย์ที่ปรึกษา รศ. ดร.ปัญญาพลด หอระตะ ที่กรุณาให้คำแนะนำสำหรับการปรึกษา ตลอดจนปรับปรุงแก้ไข ข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างดียิ่ง ผู้จัดทำขอแสดงถึงความตั้งใจจริงและความทุ่มเทของอาจารย์ และ ขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี่

ขอขอบคุณเพื่อนร่วมโครงการที่ร่วมมือกันทำงานอย่างเข้มแข็งและอดทนในการแก้ไขปัญหาที่เกิดขึ้น ตลอดจนให้กำลังใจซึ่งกันและกันจนสำเร็จ

นอกจากนี้ คณะผู้จัดทำขอขอบคุณผู้ปกครองและครอบครัวที่สนับสนุนและให้กำลังใจมาตลอดระยะเวลา ของการศึกษา

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญภาพ	ง
สารบัญตาราง	ฉ
 บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการดำเนินงาน	3
1.5 แผนและระยะเวลาในการดำเนินงาน	3
1.6 เครื่องมือที่ใช้ในการดำเนินงาน	3
1.7 ประโยชน์ที่คาดว่าจะได้รับ	4
1.8 นิยามศัพท์เฉพาะ	4
บทที่ 2 แนวคิดและทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบ	
2.1 หลักการพัฒนาซอฟต์แวร์	5
2.2 เทคโนโลยีที่เกี่ยวข้อง	7
บทที่ 3 วิธีดำเนินโครงการ	
3.1 ผลการวิเคราะห์ระบบงานปัจจุบัน	12
3.2 วิเคราะห์ความต้องการของผู้ใช้	13
3.3 วิเคราะห์ระบบงานใหม่	14
บทที่ 4 ผลการพัฒนาระบบ	
4.1 ส่วนของผู้ใช้งานทั่วไป (User)	33
4.2 ส่วนของผู้ดูแลระบบ (Admin)	49
4.3 ส่วนของผู้ขาย (Seller)	54
บทที่ 5 อภิปรายและข้อเสนอแนะ	
5.1 สรุปผลการดำเนินโครงการ	62
5.2 ปัญหาและข้อจำกัด	63
5.3 ข้อเสนอแนะ	64
บรรณานุกรม	66

สารบัญภาพ

	หน้า
ภาพที่ 1 Usecase diagram ของ Admin	14
ภาพที่ 2 Usecase diagram ของ Seller	15
ภาพที่ 3 Usecase Diagram ของ User/Buyer	16
ภาพที่ 4 Domain Model	17
ภาพที่ 5 System Sequence diagram (User ซื้อสินค้า)	18
ภาพที่ 6 System Sequence diagram (Seller เพิ่มสินค้า)	19
ภาพที่ 7 System Sequence diagram (User รีวิวสินค้า)	20
ภาพที่ 8 System Sequence diagram (Admin อนุมัติร้านค้า)	21
ภาพที่ 9 System Sequence diagram (User แซทกับ Shop)	22
ภาพที่ 10 Sequence Diagram (User ซื้อสินค้า)	23
ภาพที่ 11 Sequence Diagram (Seller เพิ่มสินค้า)	24
ภาพที่ 12 Sequence Diagram (User รีวิวสินค้า)	25
ภาพที่ 13 Sequence Diagram (Admin อนุมัติร้านค้า)	26
ภาพที่ 14 Sequence Diagram (User แซทกับ Shop)	27
ภาพที่ 15 Activity Diagram (User ซื้อสินค้า)	28
ภาพที่ 16 Activity Diagram (Seller เพิ่มสินค้า)	29
ภาพที่ 17 Activity Diagram (User รีวิวสินค้า)	30
ภาพที่ 18 Activity Diagram (Admin อนุมัติร้านค้า)	31
ภาพที่ 19 Activity Diagram (User แซทกับ Shop)	32
ภาพที่ 20 หน้าแรก (Home Page)	33
ภาพที่ 21 หน้าสมัครสมาชิกและเข้าสู่ระบบ	34
ภาพที่ 22 หน้าสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)	34
ภาพที่ 23 หน้าหลังจากการเข้าสู่ระบบ	35
ภาพที่ 24 หน้ารายการสินค้า	36
ภาพที่ 25 หน้ารายละเอียดสินค้า	37
ภาพที่ 26 หน้าตัชกร้าสินค้า	38
ภาพที่ 27 หน้าตัชกร้าสินค้า (ต่อ)	38
ภาพที่ 28 หน้าชำระเงิน	39
ภาพที่ 29 หน้าชำระเงิน (ต่อ)	40
ภาพที่ 30 หน้าชำระเงิน (ต่อ)	41
ภาพที่ 31 หน้าชำระเงิน (ต่อ)	42
ภาพที่ 32 หน้าชำระเงิน (ต่อ)	43

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 33 หน้าประวัติการสั่งซื้อ	43
ภาพที่ 34 หน้าข้อมูลส่วนตัว	44
ภาพที่ 35 หน้าตั้งค่า	44
ภาพที่ 36 หน้าคำสั่งซื้อของฉัน	45
ภาพที่ 37 หน้ารายการโปรด	45
ภาพที่ 38 หน้าແພທของຜູ້ໃຊ້	46
ภาพที่ 39 หน้าตั้งค่า	46
ภาพที่ 40 หน้าตั้งค่า (ต่อ)	47
ภาพที่ 41 หน้าເປີດຮ້ານຄ້າ	47
ภาพที่ 42 หน้าເປີດຮ້ານຄ້າ (ต่อ)	48
ภาพที่ 43 หน้าແດບອົບຂອງຜູ້ຄູແລະບບ	49
ภาพที่ 44 หน้าການຈັດກາຣູຜູ້ໃຊ້	50
ภาพที่ 45 หน้าການຈັດກາຣູນຄ້າ	51
ภาพที่ 46 หน้าການຈັດກາຣູນຄ້າ (ຕ່ອ)	51
ภาพที่ 47 หน้าການຈັດກາຣູນຄ້າ (ຕ່ອ)	52
ภาพที่ 48 หน้าການຕັ້ງຄ່າ	53
ภาพที่ 49 หน้าແດບອົບຂອງຜູ້ຂາຍ	54
ภาพที่ 50 หน้าຈັດກາສິນຄ້າ	55
ภาพที่ 51 หน้าຈັດກາຄຳສັ່ງຊື້	56
ภาพที่ 52 หน้าຮ້າຍລະເມືດຄຳສັ່ງຊື້	57
ภาพที่ 53 หน້າແພທຂອງຜູ້ຂາຍ	58
ภาพที่ 54 หน້າການຈັດກາຣູວິວ	59
ภาพที่ 55 หน້າການຈັດກາຣູນຄ້າ	60
ภาพที่ 56 หน້າການຈັດກາຣູນຄ້າ (ຕ່ອ)	61

สารบัญตาราง

ตารางที่ 1 แผนและระยะเวลาในการดำเนินงานของโครงการ	หน้า 3
---	--------

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ในยุคดิจิทัลปัจจุบัน การพาณิชย์อิเล็กทรอนิกส์ (E-commerce) เป็นส่วนสำคัญของระบบเศรษฐกิจทั่วโลก โดยเฉพาะในประเทศไทยที่ตลาดออนไลน์เติบโตอย่างรวดเร็ว ผู้บริโภคหันมาใช้บริการซื้อขายสินค้าผ่านแพลตฟอร์มออนไลน์มากขึ้น เนื่องจากความสะดวกสบาย ความรวดเร็ว และการเข้าถึงสินค้าที่หลากหลายตลอด 24 ชั่วโมง ในขณะเดียวกัน ผู้ประกอบการรายย่อยและ SMEs ก็มีโอกาสในการขยายช่องทางการขายและเข้าถึงลูกค้าได้ง่ายขึ้น ผ่านช่องทางออนไลน์

โครงการนี้มีจุดมุ่งหมายเพื่อพัฒนาเว็บไซต์ E-commerce สำหรับการศึกษาหลักการพัฒนาระบบ โดยใช้เทคโนโลยีที่ทันสมัย เช่น Next.js สำหรับส่วน Frontend และ Spring Boot สำหรับ Backend พร้อมกับฐานข้อมูล PostgreSQL, JPA, JWT, และระบบ Logging ระบบจะรองรับผู้ใช้งาน 3 บทบาทหลัก ได้แก่ ผู้ซื้อ, ผู้ขาย, และผู้ดูแลระบบ โดยมีฟังก์ชันการทำงานที่เหมาะสมตามบทบาท เช่น การเลือกซื้อสินค้า การรีวิวสินค้า การจัดการร้านค้า การตรวจสอบคำสั่งซื้อ และการแข่งระหว่างผู้ซื้อกับผู้ขาย

การจัดทำโครงการนี้จะสามารถศึกษาและฝึกฝนการใช้งานเทคโนโลยีต่าง ๆ ในการพัฒนาเว็บไซต์และระบบที่รองรับผู้ใช้หลายบทบาท เช่น CRUD, SOLID principles, การจัดการข้อมูล และการทำงานของระบบหลังบ้านและหน้าบ้าน รวมถึงการเรียนรู้การพัฒนาแอปพลิเคชันในรูปแบบที่สามารถขยายและปรับปรุงได้ในอนาคต

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาหลักการพัฒนาเว็บไซต์ E-commerce โดยใช้เทคโนโลยีที่ทันสมัย เช่น Next.js และ Spring Boot

1.2.2 เพื่อฝึกฝนการใช้งานเทคโนโลยีและเครื่องมือที่จำเป็น เช่น CRUD operations, SOLID principles, การจัดการฐานข้อมูล, การจัดการความปลอดภัย

1.2.3 เพื่อศึกษาการพัฒนาแพลตฟอร์มที่สามารถขยายได้ในอนาคต โดยสามารถปรับปรุงและเพิ่มฟีเจอร์ต่าง ๆ ได้ตามความต้องการ

1.3 ขอบเขตของโครงการ

โครงการนี้เป็นการพัฒนาแพลตฟอร์ม E-commerce แบบครบวงจร ที่รองรับการซื้อขายสินค้าออนไลน์ โดยแบ่งผู้ใช้งานออกเป็น 3 บทบาทหลัก ได้แก่ ผู้ดูแลระบบ (Admin), ผู้ขาย (Seller) และผู้ซื้อ (User)

1.3.1 พัฒนาสำหรับผู้ดูแลระบบ (Admin)

1) การจัดการผู้ใช้ (User Management)

โดยแสดงรายการผู้ใช้ทั้งหมดในระบบ พร้อมข้อมูล ชื่อ, อีเมล, เบอร์โทรศัพท์ บทบาทของผู้ใช้ (Admin, Seller, User), สถานะของผู้ใช้ (Active, Banned), วันที่สร้างบัญชีผู้ใช้ และสามารถระงับการใช้งาน (Ban) บัญชีผู้ใช้ได้

2) การจัดการร้านค้า (Shop Management)

ผู้ดูแลระบบสามารถอนุมัติหรือปฏิเสธคำขอสร้างร้านค้าจากผู้ใช้งาน และจัดการสถานะร้านค้า โดยมี 2 กรณี ได้แก่ ปิดร้านค้าชั่วคราว ซึ่งผู้ซื้อไม่สามารถเข้าถึงหน้าร้านค้าได้ แต่สามารถซื้อสินค้าที่เหลืออยู่ในระบบได้ และถอดสิทธิ์ร้านค้า จะทำการยกเลิกสิทธิ์การเป็นผู้ขาย ทำให้ไม่สามารถเข้าถึงฟังก์ชันการขายได้เลย

3) การตั้งค่าเว็บไซต์ (Website Settings)

ผู้ดูแลสามารถเปลี่ยนชื่อเว็บไซต์ และแก้ไขข้อมูลติดต่อ (อีเมล, เบอร์โทรศัพท์, ที่อยู่)

1.3.2 พัฒนาฟังก์ชันสำหรับผู้ใช้/ผู้ซื้อ (User/Buyer)

1) การซื้อขาย

ผู้ใช้สามารถเรียกดูและค้นหาสินค้า, เลือกซื้อสินค้าและเพิ่มลงในตะกร้าสินค้า, บันทึกสินค้าในการโปรด, และชำระเงิน โดยเป็น Mock Payment ที่ชี้ไปไม่เข้มต่อ กับระบบชำระเงินจริง

2) การติดต่อสื่อสาร

ผู้ใช้สามารถแชทกับร้านค้า (Chat with Shop)

3) การจัดการคำสั่งซื้อ

หลังจากการสั่งซื้อผู้ใช้สามารถตรวจสอบสถานะคำสั่งซื้อ และยืนยันการรับสินค้า

4) การจัดการบัญชี

ผู้ใช้สามารถส่งคำร้องขอเปิดร้านค้า, เปลี่ยนรหัสผ่าน, เพิ่มและจัดการที่อยู่การจัดส่ง

5) การรีวิวสินค้า

ภายหลังได้รับสินค้าผู้ใช้สามารถรีวิวและให้คะแนนสินค้าที่ซื้อแล้ว

1.3.3 พัฒนาฟังก์ชันสำหรับผู้ขาย (Seller)

1) แดชบอร์ด (Dashboard)

แสดงจำนวนสินค้าที่มีการลงขาย, จำนวนคำสั่งซื้อทั้งหมด, รายได้รวม, คะแนนรีวิวเฉลี่ย, รายได้ยอดหลัง 30 วัน, คำสั่งซื้อย้อนหลัง 30 วัน, การกระจายคะแนนรีวิว, จำนวนสินค้าที่รอดำเนินการ, จำนวนสินค้าที่จัดส่งเสร็จสิ้น, จำนวนสินค้าที่ใกล้หมดสต็อกม รายการคำสั่งซื้อล่าสุด

2) การจัดการสินค้า (Product Management)

เพิ่มสินค้าใหม่, แก้ไขข้อมูลสินค้า, ลบสินค้า, ค้นหาและกรองสินค้า

3) การจัดการคำสั่งซื้อ (Order Management)

ดูรายการคำสั่งซื้อแยกตามสถานะ, ค้นหาคำสั่งซื้อ, ยืนยันการจัดส่งสินค้า, อัปเดตสถานะคำสั่งซื้อ

4) การสื่อสารกับลูกค้า

ร้านค้าสามารถแชทกับลูกค้า (Customer Chat)

5) การจัดการรีวิว (Review Management)

ร้านค้าสามารถดูคะแนนรีวิวทั้งหมด, ดูการกระจายของคะแนน, ค้นหารีวิว และตอบกลับรีวิว

1.4 ขั้นตอนการดำเนินงาน

1.4.1 สำรวจความต้องการและรวบรวมข้อมูล

รวบรวมข้อมูลและความต้องการที่จำเป็นในการพัฒนาระบบ

1.4.2 ออกแบบระบบ

ออกแบบแพนกว่าพัฒนาตามสัมภันธ์ของข้อมูล และ Use case สำหรับระบบฐานข้อมูล และออกแบบ UI/UX สำหรับส่วนติดต่อผู้ใช้

1.4.3 พัฒนาระบบ

ใช้เทคโนโลยีที่เลือกพัฒนาเว็บไซต์และระบบต่าง ๆ ตามแบบจำลองที่ออกแบบไว้

1.4.4 ทดสอบระบบ

ทดสอบการทำงานของแต่ละฟังก์ชันเพื่อหาข้อผิดพลาดและทำการแก้ไข

1.4.5 จัดทำเอกสาร

จัดทำเอกสารที่เกี่ยวข้องกับระบบเพื่อการนำเสนอและรายงานผลการพัฒนา

1.5 แผนและระยะเวลาในการดำเนินงาน

ตารางที่ 1 แผนและระยะเวลาในการดำเนินงานของโครงการ

กิจกรรม	สิงหาคม				กันยายน				ตุลาคม			
	1	2	3	4	1	2	3	4	1	2	3	4
1. สำรวจความต้องการและรวบรวมข้อมูล												
2. ออกแบบระบบ												
3. พัฒนาระบบ												
4. ทดสอบระบบ												
5. จัดทำเอกสาร												

1.6 เครื่องมือที่ใช้ในการดำเนินงาน

1.6.1 Frontend

1) Next.js: ใช้ในการพัฒนาหน้าเว็บ

2) TypeScript: ใช้ในการเขียนโค้ดที่มีความปลอดภัยและสะดวกในการดีบัก

3) HTML, CSS: ใช้ในการออกแบบโครงสร้างหน้าเว็บ

4) Tailwind CSS: ใช้ในการออกแบบ UI ให้สวยงามและตอบสนอง

1.6.2 Backend

1) Spring Boot: ใช้ในการพัฒนา API และการจัดการข้อมูล

2) PostgreSQL: ใช้ในการจัดเก็บข้อมูล

3) JPA (Java Persistence API): ใช้ในการจัดการกับฐานข้อมูล

4) JWT (JSON Web Token): ใช้ในการยืนยันตัวตนผู้ใช้

5) CORS (Cross-Origin Resource Sharing): ใช้ในการจัดการการเข้าถึงข้อมูลจากแหล่งต่างๆ

6) Logging: ใช้ในการบันทึกข้อมูลและตรวจสอบความผิดพลาด

1.6.3 เครื่องมือที่ใช้พัฒนา

1) Visual Studio Code: ใช้ในการเขียนและจัดการโค้ด

2) Git และ GitHub: ระบบควบคุมเวอร์ชัน (Version Control System) ที่ใช้ในการจัดการโค้ดและทำงานเป็นทีม

3) Postman: เครื่องมือสำหรับทดสอบ API ที่ช่วยให้การพัฒนาและ Debug API เป็นไปอย่างมีประสิทธิภาพ

4) pgAdmin: เครื่องมือสำหรับจัดการและบริหารฐานข้อมูล PostgreSQL แบบ Graphical User Interface

1.7 ประโยชน์ที่คาดว่าจะได้รับ

1.7.1 ได้เรียนรู้กระบวนการพัฒนาโครงงานที่สามารถใช้งานได้จริงในเชิงปฏิบัติ

1.7.2 สร้างความเข้าใจในการใช้เทคโนโลยีต่าง ๆ เช่น Next.js, Spring Boot, PostgreSQL, JWT, และการพัฒนาระบบที่มีความปลอดภัย

1.7.3 เรียนรู้วิธีการออกแบบและพัฒนาระบบที่สามารถรองรับการขยายฟีเจอร์ใหม่ ๆ ในอนาคต

1.8 นิยามศัพท์เฉพาะ

1.8.1 E-commerce (พาณิชย์อิเล็กทรอนิกส์): การซื้อขายสินค้าและบริการผ่านระบบอินเทอร์เน็ตหรือเครือข่ายอิเล็กทรอนิกส์อื่นๆ โดยไม่ต้องมีการติดต่อ กันแบบพบทหน้าโดยตรง

1.8.2 Buyer (ผู้ซื้อ): ผู้ใช้งานที่เข้ามาใช้บริการแพลตฟอร์มเพื่อค้นหา เสือกซื้อ และชำระเงินสำหรับสินค้าหรือบริการต่างๆ พร้อมทั้งสามารถให้คะแนนและรีวิวสินค้าที่ซื้อได้

1.8.3 Seller (ผู้ขาย): ผู้ใช้งานที่เริ่มต้นจากการเป็น Buyer และได้เปิดร้านค้าภายใต้แพลตฟอร์ม สามารถเพิ่มสินค้า จัดการคำสั่งซื้อ ตอบกลับรีวิว และสื่อสารกับลูกค้าได้

1.8.4 Admin (ผู้ดูแลระบบ): ผู้ใช้งานที่มีสิทธิสูงสุดในการจัดการแพลตฟอร์ม สามารถควบคุมผู้ใช้งาน จัดการร้านค้า และแบบร้านค้าที่ฝ่ายนักวุฒิเบียบได้

บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบ

การพัฒนาระบบซอฟต์แวร์สมัยใหม่ต้องอาศัยแนวคิดและเทคโนโลยีหลากหลายด้านเพื่อให้ระบบมีความปลอดภัย มีประสิทธิภาพ และสามารถตอบสนองความต้องการของผู้ใช้งานได้อย่างเหมาะสม

2.1 หลักการพัฒนาซอฟต์แวร์

2.1.1 CRUD Operations

การพัฒนาแอปพลิเคชันที่มีการจัดการข้อมูลในระบบฐานข้อมูลจำเป็นต้องอาศัยหลักการพื้นฐานที่เรียกว่า CRUD Operations ซึ่งเป็นตัวย่อของ Create (สร้าง), Read (อ่าน), Update (แก้ไข) และ Delete (ลบ) หลักการนี้ถือเป็นรากฐานสำคัญในการพัฒนาซอฟต์แวร์ที่ต้องมีการจัดการข้อมูล (Martin, 2008) การดำเนินการ Create ใช้สำหรับการเพิ่มข้อมูลใหม่เข้าสู่ระบบ เช่น การสร้างบัญชีผู้ใช้ใหม่หรือการเพิ่มสินค้าใหม่ ในขณะที่การดำเนินการ Read ทำหน้าที่ในการเรียกดูและค้นหาข้อมูลที่มีอยู่ในระบบ สำหรับการแก้ไขข้อมูลจะใช้การดำเนินการ Update เช่น การปรับปรุงข้อมูลสินค้าหรือการอัปเดตสถานะคำสั่งซื้อ ส่วนการดำเนินการ Delete จะใช้สำหรับการลบข้อมูลออกจากระบบ เช่น การลบสินค้าหรือการยกเลิกคำสั่งซื้อ (Fowler, 2002)

ในการประยุกต์ใช้กับระบบพารามิเตอร์อิเล็กทรอนิกส์ หลักการ CRUD จะถูกนำมาใช้ในทุกส่วนของการจัดการข้อมูล ตั้งแต่การจัดการผู้ใช้ สินค้า คำสั่งซื้อ ร้านค้า และรีวิว โดย Spring Boot Framework จะจัดการ CRUD Operations ผ่านทาง JPA Repository และ RESTful API ในขณะที่ส่วน Frontend จะเรียกใช้งานผ่าน HTTP Methods ได้แก่ POST สำหรับ Create, GET สำหรับ Read, PUT หรือ PATCH สำหรับ Update และ DELETE สำหรับการลบข้อมูล (Walls, 2016)

2.1.2 SOLID Principles

SOLID Principles เป็นหลักการออกแบบซอฟต์แวร์ที่ประกอบด้วย 5 หลักการหลัก ซึ่งช่วยให้โค้ดมีความยืดหยุ่น ง่ายต่อการบำรุงรักษา และสามารถขยายได้ หลักการเหล่านี้ได้รับการเสนอโดย Robert C. Martin เพื่อส่งเสริมการออกแบบซอฟต์แวร์ที่มีคุณภาพ (Martin, 2000)

1) Single Responsibility Principle (SRP)

ระบุว่าคลาสหนึ่งควรมีหน้าที่รับผิดชอบเพียงอย่างเดียว และมีเหตุผลเดียวที่จะต้องเปลี่ยนแปลง การประยุกต์ใช้หลักการนี้ทำได้โดยการแยก Service Classes ตามหน้าที่ เช่น UserService, ProductService และ OrderService รวมถึงการแยก Controller สำหรับแต่ละ Domain และการแยก Repository สำหรับแต่ละ Entity (Martin, 2003)

2) Open/Closed Principle (OCP)

กำหนดว่าซอฟต์แวร์ควรเปิดให้ขยายได้แต่ปิดไม่ให้แก้ไข ในการพัฒนาระบบจะใช้ Interface สำหรับ Service Layer ทำให้สามารถสร้าง Implementation ใหม่ได้โดยไม่ต้องแก้ไขโค้ดเดิม รวมถึงการใช้ Strategy Pattern สำหรับระบบการชำระเงินที่อาจมีหลายวิธี (Gamma et al., 1994)

3) Liskov Substitution Principle (LSP)

ระบุว่า Object ของ Subclass ควรสามารถแทนที่ Object ของ Superclass ได้โดยไม่ทำให้โปรแกรมทำงานผิดพลาด การออกแบบ Class Hierarchy ของ User ที่มี Admin, Seller และ Buyer เป็น Subclass จึงต้องปฏิบัติตามหลักการนี้อย่างเคร่งครัด (Liskov, 1987)

4) Interface Segregation Principle (ISP)

กำหนดว่าไม่ควรบังคับให้ Client ต้อง Depend กับ Interface ที่มันไม่ได้ใช้ ดังนั้นควรแยก Interface สำหรับแต่ละ Role เช่น IAdminService, ISellerService และ IUserService แทนการใช้ Interface ขนาดใหญ่ที่มีทุกอย่าง (Martin, 2002)

5) Dependency Inversion Principle (DIP)

ระบุว่า Module ระดับสูงไม่ควร Depend กับ Module ระดับต่ำ ทั้งสองควร Depend กับ Abstraction การใช้ Dependency Injection ของ Spring Boot จึงเป็นการประยุกต์ใช้หลักการนี้ โดย Controller จะ Depend กับ Service Interface ไม่ใช่ Concrete Class (Fowler, 2004)

2.1.3 RESTful API Design

REST (Representational State Transfer) เป็น Architectural Style สำหรับออกแบบ Web Services ที่ใช้ HTTP Protocol ได้รับการเสนอโดย Roy Fielding ในปี 2000 ซึ่งกำหนดหลักการพื้นฐานที่สำคัญ หลายประการ (Fielding, 2000) หลักการแรกคือ Client-Server Architecture ที่แยกส่วน Frontend และ Backend อย่างชัดเจน ทำให้สามารถพัฒนาแต่ละส่วนได้อิสระ หลักการที่สองคือ Stateless ซึ่งแต่ละ Request ต้องมีข้อมูลครบถ้วนเพื่อการประมวลผล และ Server ไม่เก็บ Session State หลักการที่สามคือ Cacheable ที่ Response ควรระบุว่าสามารถ Cache ได้หรือไม่ เพื่อเพิ่มประสิทธิภาพของระบบ (Richardson & Ruby, 2007) หลักการที่สี่คือ Uniform Interface ที่ใช้มาตรฐานเดียวกันในการออกแบบ API โดยใช้ HTTP Methods มาตรฐาน ได้แก่ GET สำหรับดึงข้อมูล (Read), POST สำหรับสร้างข้อมูลใหม่ (Create), PUT หรือ PATCH สำหรับแก้ไขข้อมูล (Update) และ DELETE สำหรับลบข้อมูล (Delete) หลักการสุดท้ายคือ Layered System ที่สามารถมี Intermediate Server เช่น Load Balancer หรือ Cache Server เพื่อเพิ่มความสามารถในการขยายระบบ (Fielding & Taylor, 2002)

ในการออกแบบ API Endpoints ควรปฏิบัติตามมาตรฐาน RESTful โดยใช้ Resource-based URL เช่น GET /api/products สำหรับดูรายการสินค้าทั้งหมด, GET /api/products/{id} สำหรับดูรายละเอียดสินค้า, POST /api/products สำหรับเพิ่มสินค้าใหม่, PUT /api/products/{id} สำหรับแก้ไขข้อมูลสินค้า และ DELETE /api/products/{id} สำหรับลบสินค้า การออกแบบแบบนี้ทำให้ API มีความชัดเจน เข้าใจง่าย และ สอดคล้องกับมาตรฐานสากล (Masse, 2011)

2.1.4 MVC Architecture Pattern

MVC (Model-View-Controller) เป็น Architectural Pattern ที่แยกแอปพลิเคชันออกเป็น 3 ส่วนหลัก เพื่อลดความซับซ้อนและเพิ่มความยืดหยุ่นในการพัฒนา Pattern นี้ได้รับการพัฒนาครั้งแรกโดย Trygve Reenskaug ในช่วงปลายทศวรรษ 1970 (Reenskaug, 1979)

1) Model ที่ทำหน้าที่จัดการข้อมูลและ Business Logic ประกอบด้วย Entity Classes เช่น Product, User และ Order รวมถึง Repository Layer และ Service Layer

2) View ที่แสดงผลข้อมูลให้ผู้ใช้เห็น ซึ่งในโปรเจคนี้จะเป็น Next.js Components, React UI Components และ HTML/CSS/Tailwind

3) Controller ที่รับ Input จากผู้ใช้และควบคุม Flow ประกอบด้วย Spring Boot Controllers, API Endpoints และ Request Handlers (Fowler, 2002)

ในระบบที่พัฒนาด้วย Spring Boot จะใช้ MVC Pattern โดย Controller รับ HTTP Request จากผู้ใช้, Service ประมวลผล Business Logic และตรวจสอบความถูกต้องของข้อมูล, และ Repository จัดการการเข้าถึงฐานข้อมูล สำหรับส่วน Frontend ที่พัฒนาด้วย Next.js จะใช้ Component-based Architecture ที่มีแนวคิดคล้ายกับ MVC โดย Components ทำหน้าที่เป็น View, API Calls เป็น Controller และ State Management เป็น Model การแยกส่วนในลักษณะนี้ทำให้โค้ดมีการจัดระเบียบที่ดี ง่ายต่อการบำรุงรักษา และสามารถทำงานเป็นทีมได้อย่างมีประสิทธิภาพ (Leff & Rayfield, 2001)

2.2 เทคโนโลยีที่เกี่ยวข้อง

2.2.1 Frontend Technologies

2.2.1.1 Next.js

Next.js เป็น React Framework ที่พัฒนาโดย Vercel มีคุณสมบัติที่ช่วยในการพัฒนา Web Application สมัยใหม่อย่างมีประสิทธิภาพ คุณสมบัติสำคัญประการแรกคือ Server-Side Rendering (SSR) ซึ่งเป็นการ Render หน้าเว็บที่ Server ก่อนส่งไปยัง Client ทำให้หน้าเว็บโหลดเร็วขึ้น มี SEO ที่ดีขึ้น และประสบการณ์ผู้ใช้ที่ดีขึ้น (Vercel, 2021) คุณสมบัติที่สองคือ Static Site Generation (SSG) ที่เป็นการสร้างหน้าเว็บ Static ไว้ล่วงหน้า เหมาะกับหน้าที่ข้อมูลไม่เปลี่ยนแปลงบ่อยและต้องการความเร็วในการโหลดสูงสุด คุณสมบัติที่สามคือ File-based Routing ที่สร้าง Route อัตโนมัติจาก File Structure โดยไม่ต้องตั้งค่า Router เองทำให้โครงสร้างโปรเจกซัดเจนและจัดการได้ง่าย คุณสมบัติที่สี่คือ API Routes ที่สามารถสร้าง API Endpoints ใน Next.js ได้โดยตรง หมายเหตุการพัฒนา Serverless Functions แม้ว่าในโปรเจคนี้จะใช้ Spring Boot เป็น Backend หลัก นอกจากนี้ยังมี Image Optimization ที่ปรับขนาดและคุณภาพภาพอัตโนมัติเพื่อเพิ่มประสิทธิภาพการโหลดหน้าเว็บ (Next.js Documentation, 2023)

ในการประยุกต์ใช้กับโปรเจกจะใช้ Next.js สำหรับสร้าง Pages สำหรับแต่ละ Route เช่น Home, Products, Cart และ Admin Dashboard

2.2.1.2 TypeScript

TypeScript เป็น Superset ของ JavaScript ที่เพิ่ม Type System เข้ามา พัฒนาโดย Microsoft เพื่อแก้ไขปัญหาของ JavaScript ที่เป็น Dynamically Typed Language (Microsoft, 2012) ข้อดีที่สำคัญของ TypeScript คือ Type Safety ที่ช่วยตรวจสอบข้อผิดพลาดตั้งแต่เวลาเขียนโค้ด ก่อนที่จะรันโปรแกรม ทำให้ลดโอกาสเกิด Runtime Error ได้อย่างมาก นอกจากนี้ยังมี Better IDE Support ที่ให้ Autocomplete และ IntelliSense ที่ดีขึ้น ช่วยเพิ่มประสิทธิภาพในการเขียนโค้ด (Bierman et al., 2014)

ข้อดีอีก 1 ของ TypeScript ได้แก่ Refactoring ที่ง่ายและปลอดภัยขึ้นเนื่องจากมี Type Checking, Documentation ที่ดีขึ้น เพราะ Type เป็น Documentation ในตัว และ Maintainability ที่ทำ

ให้โค้ดอ่านง่ายและบำรุงรักษาได้ดีขึ้น โดยเฉพาะในโปรเจคขนาดใหญ่ที่มีหลายคนร่วมพัฒนา ในการประยุกต์ใช้กับโปรเจคจะสร้าง Interface สำหรับทุก Entity เช่น Product, User และ Order เพื่อกำหนดโครงสร้างข้อมูลอย่างชัดเจน รวมถึงการสร้าง Type Definitions สำหรับ API Response และ Request, การใช้ Type Guards สำหรับ Validation และการสร้าง Enum สำหรับค่าคงที่ เช่น OrderStatus และ UserRole (Cherny, 2019)

2.2.1.3 Tailwind CSS

Tailwind CSS เป็น Utility-first CSS Framework ที่ให้เขียน CSS ผ่าน Utility Class Names แทนการเขียน CSS แบบเดิม คุณสมบัติเด่นของ Tailwind คือการใช้ Utility-First Approach ที่สามารถสร้าง UI ได้รวดเร็วโดยใช้ Class Names ที่กำหนดไว้แล้ว เช่น bg-blue-500 สำหรับสีพื้นหลังสีน้ำเงิน หรือ py-2 px-4 สำหรับ Padding (Tailwind Labs, 2019) นอกจากนี้ยังรองรับ Responsive Design อย่างง่ายดาย ผ่าน Breakpoints ที่กำหนดไว้ เช่น md:w-1/2 สำหรับความกว้าง 50% บนหน้าจอขนาดกลาง

คุณสมบัติที่สำคัญอีกประการหนึ่งคือ JIT (Just-In-Time) Mode ที่ Compile เข้าไปในไฟล์ CSS ที่ใช้จริง ทำให้ไฟล์ CSS มีขนาดเล็กและประสิทธิภาพสูง Tailwind CSS ยังรองรับ Dark Mode ในตัวและสามารถ Customize ได้ผ่าน Configuration File ข้อดีของ Tailwind ได้แก่ การพัฒนาที่รวดเร็วขึ้น, ไฟล์ CSS ที่มีขนาดเล็ก, การออกแบบที่สอดคล้องกัน (Consistent Design) และไม่ต้องตั้งชื่อ Class ที่ซับซ้อน (Adam, 2020)

ในโปรเจคนี้จะใช้ Tailwind CSS สำหรับออกแบบ UI Components ทั้งหมด สร้าง Responsive Layout สำหรับทั้ง Mobile และ Desktop, กำหนด Custom Theme สำหรับ Brand Colors และสร้าง Reusable Components ที่สามารถกลับมาใช้ใหม่ได้

2.2.2 Backend Technologies

2.2.2.1 Spring Boot

Spring Boot เป็น Framework สำหรับการพัฒนา Java Application ที่ช่วยให้การพัฒนาเป็นไปอย่างรวดเร็วและง่ายดาย พัฒนาโดย Pivotal Team เพื่อลดความซับซ้อนในการตั้งค่า Spring Framework แบบดั้งเดิม (Walls, 2016) คุณสมบัติเด่นของ Spring Boot ประกอบด้วย Auto-Configuration ที่ตั้งค่าอัตโนมัติตาม Dependencies ที่ระบุ, Embedded Server ที่มี Tomcat หรือ Jetty ในตัวไม่ต้อง Deploy แยก, Production-Ready Features ที่มี Health Check, Metrics และ Monitoring ในตัว, Dependency Injection ที่จัดการ Dependencies อัตโนมัติ และ Spring Data JPA ที่ช่วยลดโค้ดในการจัดการ Database (Gupta, 2016)

สถาปัตยกรรมของ Spring Boot แบ่งออกเป็น 4 Layer หลัก ได้แก่ Controller Layer ที่รับ HTTP Requests จาก Client และส่งกลับ Response, Service Layer ที่ประมวลผล Business Logic และควบคุมการทำงานของระบบ, Repository Layer ที่จัดการการเข้าถึงฐานข้อมูลผ่าน JPA และ Entity Layer ที่เป็น Database Models ที่ Mapping กับตารางในฐานข้อมูล การแบ่ง Layer ในลักษณะนี้สอดคล้องกับหลักการ Separation of Concerns และทำให้โค้ดมีการจัดระเบียบที่ดี ง่ายต่อการทดสอบและบำรุงรักษา (Johnson et al., 2009)

ในโปรเจคนี้ใช้ Spring Boot สำหรับสร้าง RESTful APIs สำหรับทุกฟีเจอร์, จัดการ Authentication และ Authorization ด้วย JWT, ทำ Validation ข้อมูลก่อนบันทึกลงฐานข้อมูล, จัดการ Exception Handling อย่างเป็นระบบ และทำ Logging เพื่อติดตามการทำงานของระบบ

2.2.2.2 PostgreSQL

PostgreSQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) แบบ Open Source ที่มีความสามารถสูงและได้รับความนิยมอย่างแพร่หลาย คุณสมบัติที่สำคัญที่สุดคือการเป็น ACID Compliant ที่รับประกัน Data Integrity ผ่าน 4 หลักการ ได้แก่ Atomicity ที่ Transaction ต้องเสร็จทั้งหมดหรือไม่เสร็จเลย, Consistency ที่ข้อมูลต้องสอดคล้องกันเสมอ, Isolation ที่ Transaction แต่ละตัวแยกกัน และ Durability ที่ข้อมูลไม่สูญหายหลังจาก Commit (Stonebraker & Rowe, 1986)

PostgreSQL มี Advanced Features มากราย เช่น JSON/JSONB Support สำหรับจัดเก็บข้อมูลแบบ Document, Full-Text Search สำหรับการค้นหาข้อความ, Array และ Custom Types ที่เพิ่มความยืดหยุ่น, Window Functions สำหรับการวิเคราะห์ข้อมูล และ Common Table Expressions (CTEs) สำหรับ Query ที่ซับซ้อน (Obe & Hsu, 2014) นอกจากนี้ PostgreSQL ยังมีความสามารถใน Scalability ที่รองรับข้อมูลขนาดใหญ่, Extensibility ที่สามารถเพิ่ม Functions และ Data Types ได้ และ Reliability ที่มีความเสถียรสูง

2.2.2.3 JPA (Java Persistence API)

JPA เป็น Specification สำหรับ Object-Relational Mapping (ORM) ใน Java ที่ช่วยให้นักพัฒนาสามารถทำงานกับฐานข้อมูลผ่าน Object แทนการเขียน SQL โดยตรง หลักการทำงานของ JPA ประกอบด้วย Entity Mapping ที่แปลง Java Class เป็น Database Table, Relationship Mapping ที่จัดการความสัมพันธ์ระหว่าง Entities เช่น @OneToOne, @ManyToOne และ @ManyToMany, Query Generation ที่สร้าง SQL อัตโนมัติจากการเรียกใช้ Method และ Caching ที่เพิ่มประสิทธิภาพด้วย Cache (Keith & Stafford, 2009)

JPA ยังรองรับ JPQL (Java Persistence Query Language) ที่เป็นภาษา Query แบบ Object-oriented ทำให้สามารถเขียน Query ที่ซับซ้อนได้โดยอิสระจากฐานข้อมูลที่ใช้ ข้อดีของ JPA ได้แก่ การลดโค้ด Boilerplate ที่ซ้ำซ้อน, Database Independence ที่สามารถเปลี่ยน Database ได้ง่าย, Type-Safe Queries ที่ตรวจสอบข้อผิดพลาดตั้งแต่ Compile Time, Automatic Schema Generation ที่สร้างตารางอัตโนมัติ จาก Entity และ Transaction Management ที่จัดการ Transaction อย่างมีประสิทธิภาพ (Bauer & King, 2007)

2.2.2.4 JWT (JSON Web Token)

เป็นมาตรฐานที่ถูกกำหนดโดย Internet Engineering Task Force (IETF) ผ่าน RFC 7519 โดยเป็นวิธีการที่จะตัดรัดและปลอดภัยในการแสดงข้อมูล (claims) ที่จะถูกส่งระหว่างสองฝ่าย (Jones et al., 2015) JWT ได้รับการนำมาใช้อย่างแพร่หลายในปัจจุบันสำหรับการพิสูจน์ตัวตนและการอนุญาต (authorization) โดยเฉพาะในกรอบงาน OAuth และ OpenID (Mishra & Kalra, 2019)

โครงสร้างของ JWT ประกอบด้วยสามส่วนหลักที่แยกด้วยเครื่องหมายจุด (.) คือ Header, Payload และ Signature ส่วน Header จะเก็บข้อมูลเกี่ยวกับ Algorithm ที่ใช้ในการเข้ารหัส เช่น HS256 หรือ RS256 ส่วน Payload จะเก็บข้อมูลที่ต้องการส่ง หรือที่เรียกว่า Claims ซึ่งอาจประกอบด้วยข้อมูลผู้ใช้ เช่น userId, email, role และเวลาหมดอายุ ส่วนสุดท้ายคือ Signature ซึ่งเป็นลายเซ็นดิจิทัลที่ใช้ในการตรวจสอบความถูกต้องของ Token โดยใช้วิธีการเช่น HMAC SHA-256

กระบวนการ JWT Authentication เริ่มต้นจากการที่ผู้ใช้ส่ง Username และ Password เข้ามา จากนั้น Server จะทำการตรวจสอบข้อมูลและสร้าง JWT ส่งกลับไปให้ Client JWT จะถูกสร้างขึ้นเมื่อผู้ใช้ทำการพิสูจน์ตัวตนกับ Web Application และเมื่อได้รับ Token แล้ว Client จะแนบ Token นี้เข้ากับทุก HTTP Request ที่ส่งไปยัง Server ในลำดับถัดไป (Vaadata, 2025)

เมื่อ Client ต้องการเข้าถึง Resource ที่ต้องการการพิสูจน์ตัวตน จะส่ง JWT ผ่าน HTTP Header โดยใช้รูปแบบ "Authorization: Bearer <token>" Server จะทำการตรวจสอบความถูกต้องของ Token ด้วยการตรวจสอบลายเซ็นดิจิทัลและความถูกต้องของข้อมูลภายใน จากนั้นจึงดำเนินการตาม Request ที่ได้รับ

JWT มีข้อดีหลายประการที่ทำให้เป็นที่นิยมในการพัฒนาระบบสมัยใหม่ ประการแรกคือ การเป็น Stateless ทำให้ Server ไม่ต้องเก็บ Session ไว้ในฝั่งของตัวเอง ซึ่งช่วยลดภาระของ Server และทำให้ระบบมีความสามารถในการขยายตัว (Scalability) ได้ดี โดยเฉพาะอย่างยิ่งเมื่อมาใช้กับสถาปัตยกรรม Microservices นอกจากนี้ JWT ยังสามารถใช้งานได้ข้าม Domain (Cross-Domain) และเป็น Self-Contained คือมีข้อมูลครบในตัว Token เอง

JWT สามารถใช้ในการแลกเปลี่ยนข้อมูลระหว่างฝ่ายต่างๆ ได้อย่างปลอดภัย เนื่องจากสามารถถลงนามด้วยคุณูปแบบสารณะ/ส่วนตัว ทำให้สามารถยืนยันตัวตนของผู้ส่งได้ และเนื่องจากลายเซ็นถูกคำนวณโดยใช้ Header และ Payload จึงสามารถตรวจสอบได้ว่าเนื้อหาไม่ถูกตัดแปลง (JWT.io, 2024)

2.2.2.5 CORS (Cross-Origin Resource Sharing)

Cross-Origin Resource Sharing (CORS) เป็นกลไกที่ใช้ HTTP Headers เพื่ออนุญาตให้ Server สามารถรับ Origins (domain, scheme หรือ port) อื่นๆ นอกเหนือจาก Origin ของตัวเองที่ Browser ควรอนุญาตให้โหลด Resources ได้ (MDN Web Docs, 2025) CORS เป็นกลไกความปลอดภัยที่ช่วยแก้ไขข้อจำกัดของ Same-Origin Policy อย่างปลอดภัย โดยอนุญาตให้ Web Page สามารถเข้าถึง Resources ที่ถูกจำกัดจาก Server ที่อยู่บน Domain ที่แตกต่างจาก Domain ที่ให้บริการ Web Page นั้น (Wikipedia, 2025)

ปัญหาหลักที่ CORS ช่วยแก้ไขคือ Browser จะบล็อกการเรียก API ข้าม Domain ตามนโยบาย Same-Origin Policy เพื่อความปลอดภัย ซึ่งในอดีต XMLHttpRequest (XHR) Level 1 API อนุญาตให้ส่ง Request ได้เฉพาะภายใน Origin เดียวเท่านั้น (OWASP, 2025) โดยนโยบาย Same-Origin Policy นี้แม้จะมีความปลอดภัยสูง แต่ก็ขาดความยืดหยุ่นสำหรับกรณีการใช้งานจริงที่ถูกต้องตามกฎหมาย (AWS, 2025)

CORS ทำงานผ่านชุดของ HTTP Headers ที่กำหนด Web Origins ที่เชื่อมต่อได้และคุณสมบัติที่เกี่ยวข้อง เช่น การอนุญาตให้เข้าถึงแบบมีการพิสูจน์ตัวตน (authenticated access) โดย Headers เหล่านี้จะถูกรวบกันในการแลกเปลี่ยนระหว่าง Browser และ Cross-Origin Website ที่พยายามจะเข้าถึง (PortSwiggy, 2025) Browser จะทำการส่ง "Preflight" Request ไปยัง Server เพื่อตรวจสอบว่า Server อนุญาตให้ส่ง Request จริงหรือไม่ก่อนที่จะส่ง Request จริง (MDN Web Docs, 2025)

แม้ CORS จะช่วยให้ Web Applications สามารถทำงานร่วมกับ Third-party APIs และฝัง Content จาก Sites อื่นได้ แต่หากไม่ได้รับการตั้งค่าอย่างถูกต้องก็อาจก่อให้เกิดความเสี่ยงด้านความปลอดภัย (Suryawanshi, 2023) ซึ่งหนึ่งของ CORS เกิดขึ้นเมื่อ Web Application อนุญาต Cross-Domain Requests โดยไม่มีการตรวจสอบ Origin อย่างเหมาะสม ซึ่งอาจนำไปสู่การโจมตีจากผู้ไม่หวังดี (Suryawanshi, 2023)

การตั้งค่า CORS จำเป็นต้องมีการพิจารณาด้านความปลอดภัยอย่างรอบคอบ เพื่อหลีกเลี่ยงช่องโหว่ต่างๆ ของ Web Application (Tenable, 2023) ในทางปฏิบัติ ควรระบุ Origins ที่เชื่อถือได้อย่างชัดเจนแทนการใช้ Wildcard (*) และควรจำกัด Methods และ Headers เฉพาะที่จำเป็นต่อการทำงานของระบบเท่านั้น

บทที่ 3

วิธีดำเนินโครงการ

3.1 ผลการวิเคราะห์ระบบงานปัจจุบัน

ในการวิเคราะห์ระบบงานปัจจุบันของเว็บไซต์พานิชย์อิเล็กทรอนิกส์ที่มีชื่อเสียงอย่าง Amazon หรือ Shopee เราสามารถสังเกตข้อดีและข้อจำกัดของระบบเหล่านี้เพื่อเป็นแนวทางในการพัฒนาระบบที่มีประสิทธิภาพยิ่งขึ้น

เว็บไซต์พานิชย์อิเล็กทรอนิกส์ชั้นนำเหล่านี้ได้รับการออกแบบมาอย่างดีเยี่ยมเพื่อให้ผู้ใช้สามารถเข้าถึงข้อมูลได้อย่างสะดวกและรวดเร็ว โดยเฉพาะการค้นหาสินค้าที่ผู้ใช้ต้องการ ซึ่งระบบการค้นหานั้นมีความแม่นยำและตอบสนองได้อย่างรวดเร็ว นอกจากนี้ยังมีฟีเจอร์การชำระเงินที่หลากหลายและปลอดภัย รวมถึงระบบติดตามคำสั่งซื้อที่ช่วยให้ผู้ซื้อสามารถทราบสถานะการจัดส่งได้ทันที ถือเป็นการเพิ่มประสบการณ์การใช้งานที่น่าพึงพอใจให้กับผู้ใช้ทุกคน

นอกจากนี้ ระบบเหล่านี้ยังให้ความสำคัญกับการสนับสนุนผู้ใช้ โดยการมีฟีเจอร์เช่น การรีวิวสินค้า ที่ช่วยให้ผู้ซื้อสามารถตัดสินใจซื้อสินค้าจากความคิดเห็นของผู้ใช้รายอื่นได้อย่างมั่นใจ ระบบแนะนำสินค้าที่ใช้เทคโนโลยี AI ยังช่วยเพิ่มโอกาสในการซื้อสินค้าที่ตรงกับความสนใจของผู้ใช้มากขึ้น รวมทั้งการให้บริการช่องทางสนับสนุนผ่านการแชทออนไลน์ ซึ่งทำให้ผู้ซื้อสามารถติดต่อ กับผู้ขายหรือฝ่ายบริการลูกค้าได้ตลอดเวลา

อย่างไรก็ตาม ระบบเหล่านี้ก็ยังมีข้อจำกัดที่ผู้ใช้งานบางรายอาจพบเจอ โดยเฉพาะผู้ขายบางรายที่อาจรู้สึกว่าการใช้งานระบบค่อนข้างซับซ้อน ไม่ว่าจะเป็นการเพิ่มสินค้าหรือการจัดการข้อมูลต่างๆ ที่อาจไม่มีเครื่องมือที่ช่วยให้การทำงานเป็นไปอย่างราบรื่นและสะดวกเท่าที่ควร

นอกจากนี้ ฟีเจอร์การแนะนำสินค้าที่ระบบบางแห่งใช้ อาจยังไม่สามารถปรับแต่งให้ตรงกับความต้องการของผู้ใช้แต่ละคนได้อย่างละเอียด ซึ่งทำให้การแนะนำสินค้าบางครั้งอาจไม่แม่นยำหรือไม่สอดคล้องกับความสนใจของผู้ใช้

อีกหนึ่งข้อจำกัดที่พบได้ในระบบปัจจุบันคือ ระบบการแชทระหว่างผู้ซื้อและผู้ขาย ที่บางครั้งอาจไม่ได้รับการตอบสนองทันทีเมื่อไป อาจทำให้ผู้ซื้อรู้สึกไม่สะดวกหากมีคำถามหรือข้อสงสัยเกี่ยวกับสินค้าและบริการ

จากข้อดีและข้อจำกัดของระบบงานปัจจุบันที่กล่าวถึง เราสามารถนำมาเป็นแนวทางในการพัฒนาระบบที่มีให้ตอบสนองความต้องการของผู้ใช้ได้ดียิ่งขึ้น โดยมุ่งเน้นไปที่การเพิ่มประสิทธิภาพในการใช้งาน การพัฒนาฟีเจอร์ที่เหมาะสมกับแต่ละบทบาทของผู้ใช้ และการเสริมสร้างความสะดวกสบายในการใช้งาน เพื่อให้ผู้ใช้สามารถมีประสบการณ์ที่ดีที่สุดในการซื้อขายออนไลน์

3.2 วิเคราะห์ความต้องการของผู้ใช้

3.2.1 ความต้องการของผู้ซื้อ

1) ผู้ซื้อมักต้องการระบบที่สามารถค้นหาสินค้าได้อย่างรวดเร็ว พร้อมกับฟีเจอร์กรองสินค้าที่สะดวก เช่น การกรองตามประเภท ราคา หรือรีวิว

2) ระบบควรมีหลายช่องทางให้ผู้ซื้อเลือกในการชำระเงิน เช่น บัตรเครดิต, การโอนเงิน, หรือ การชำระผ่านช่องทางดิจิทัล

3) ผู้ซื้อต้องการสามารถตรวจสอบสถานการณ์จัดส่งสินค้าและติดตามสถานการณ์ชำระเงินได้

3.2.2 ความต้องการของผู้ขาย

1) ผู้ขายต้องการเครื่องมือที่ช่วยในการเพิ่มสินค้าหรือจัดการกับข้อมูลสินค้าต่างๆ เช่น การอัปโหลดรูปภาพสินค้า การตั้งราคา

2) การตรวจสอบคำสั่งซื้อ ผู้ขายต้องการสามารถตรวจสอบสถานะของคำสั่งซื้อ และจัดการคำสั่งซื้อที่เกิดขึ้นในระบบได้

3) การตอบรับและจัดการรีวิว: ผู้ขายต้องการระบบที่สามารถติดตามและตอบกลับรีวิวจากลูกค้าได้ง่าย

3.2.3 ความต้องการของผู้ดูแลระบบ

1) ผู้ดูแลระบบต้องสามารถดูแลและตรวจสอบบัญชีผู้ใช้ รวมถึงการตรวจสอบกิจกรรมที่เกิดขึ้นในระบบ

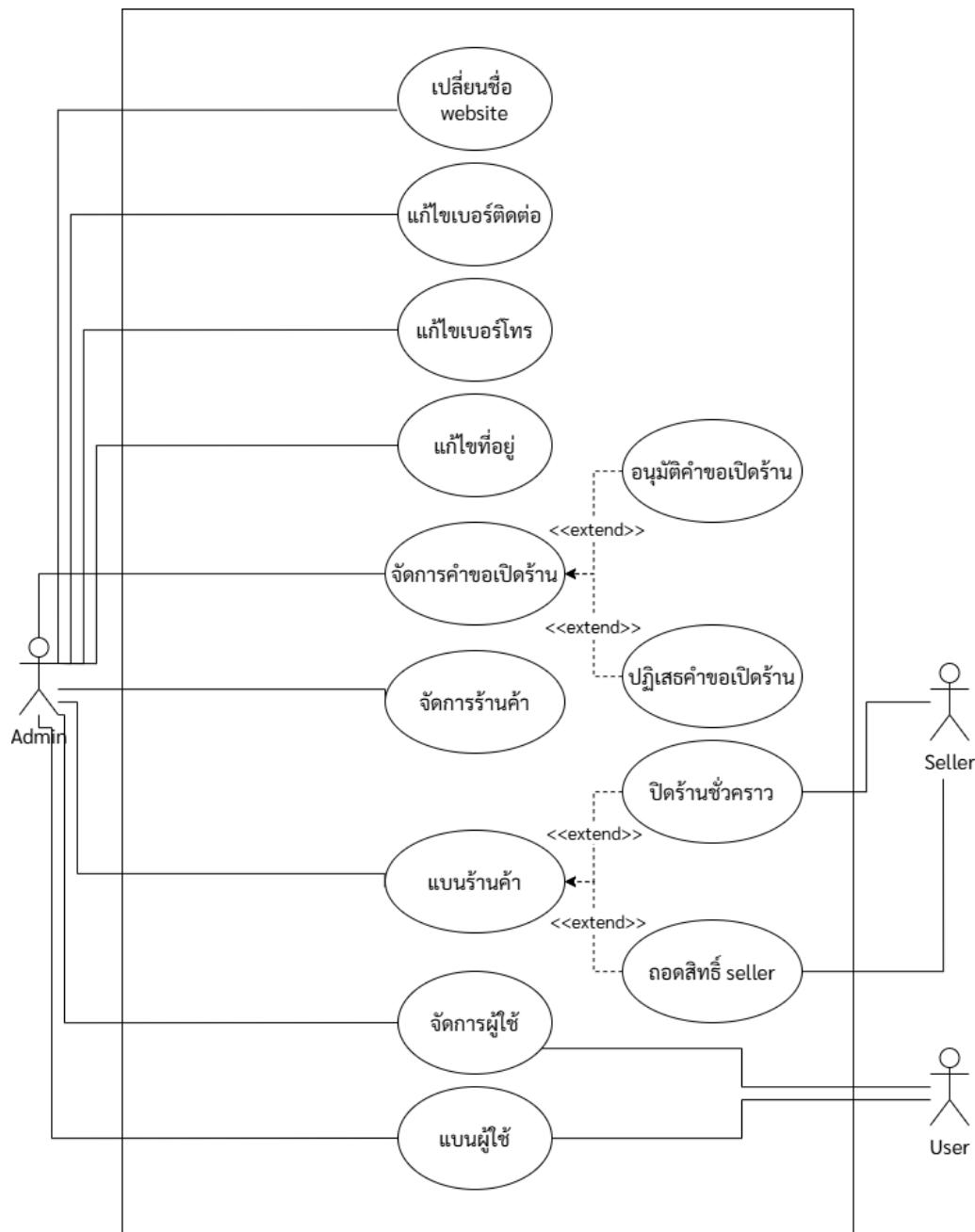
2) ผู้ดูแลระบบต้องสามารถແນผู้ขายหรือร้านค้าที่ฝ่าฝืนกฎระเบียบของแพลตฟอร์มได้

3.3 วิเคราะห์ระบบงานใหม่

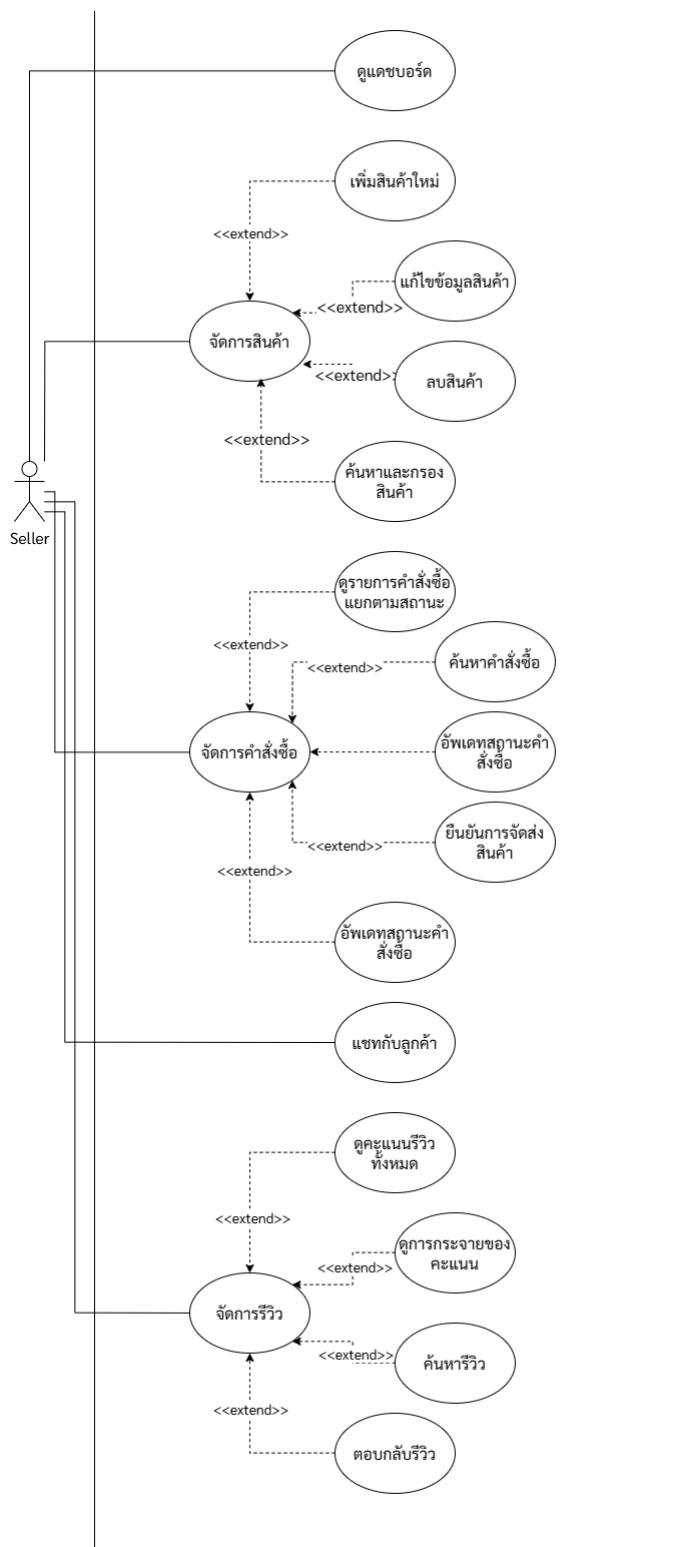
การพัฒนาระบบใหม่ของเว็บไซต์พาณิชย์อิเล็กทรอนิกส์ของเราต้องมุ่งเน้นการตอบสนองต่อความต้องการของผู้ใช้ โดยมีการออกแบบระบบได้ดังนี้

3.3.1 Usecase Diagram

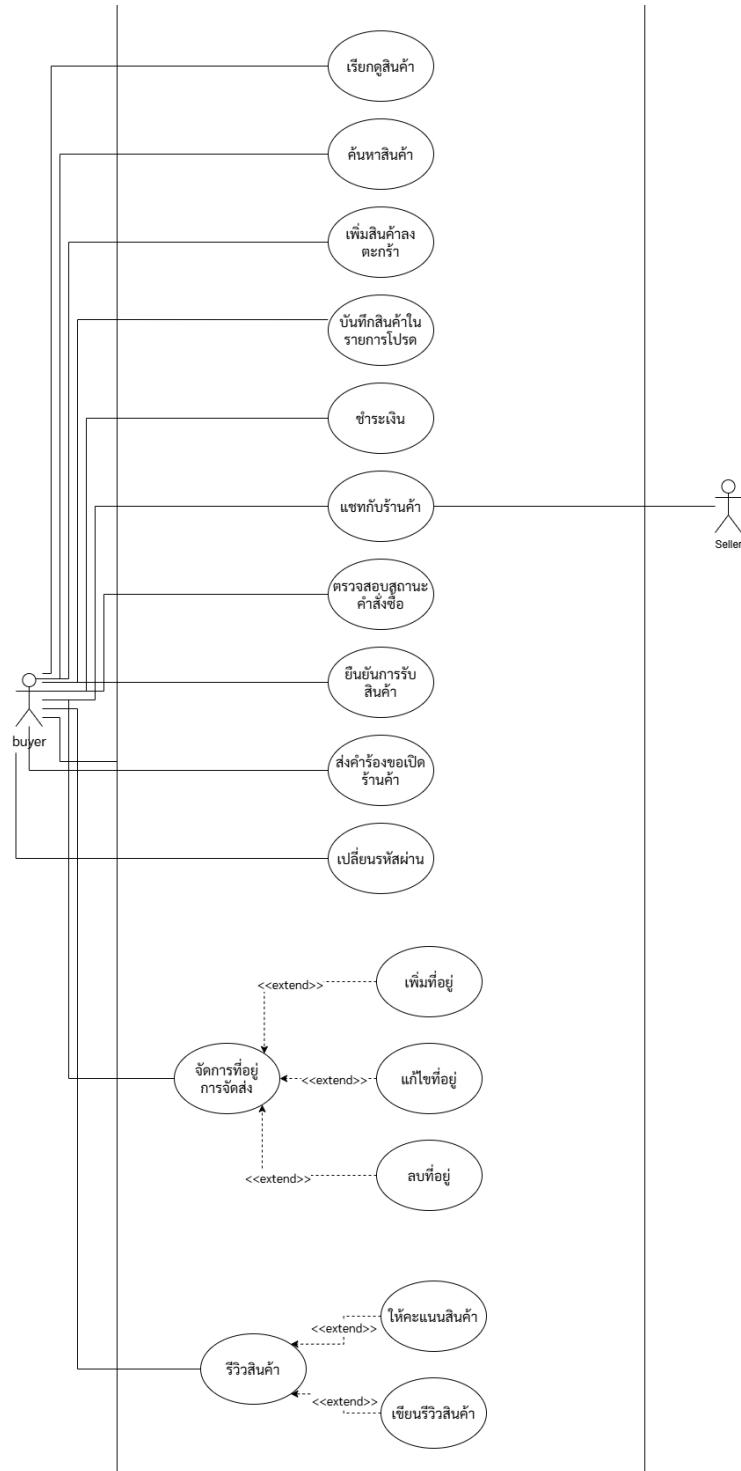
ภาพที่ 1 Usecase diagram ของ Admin



ภาพที่ 2 Usecase diagram ของ Seller

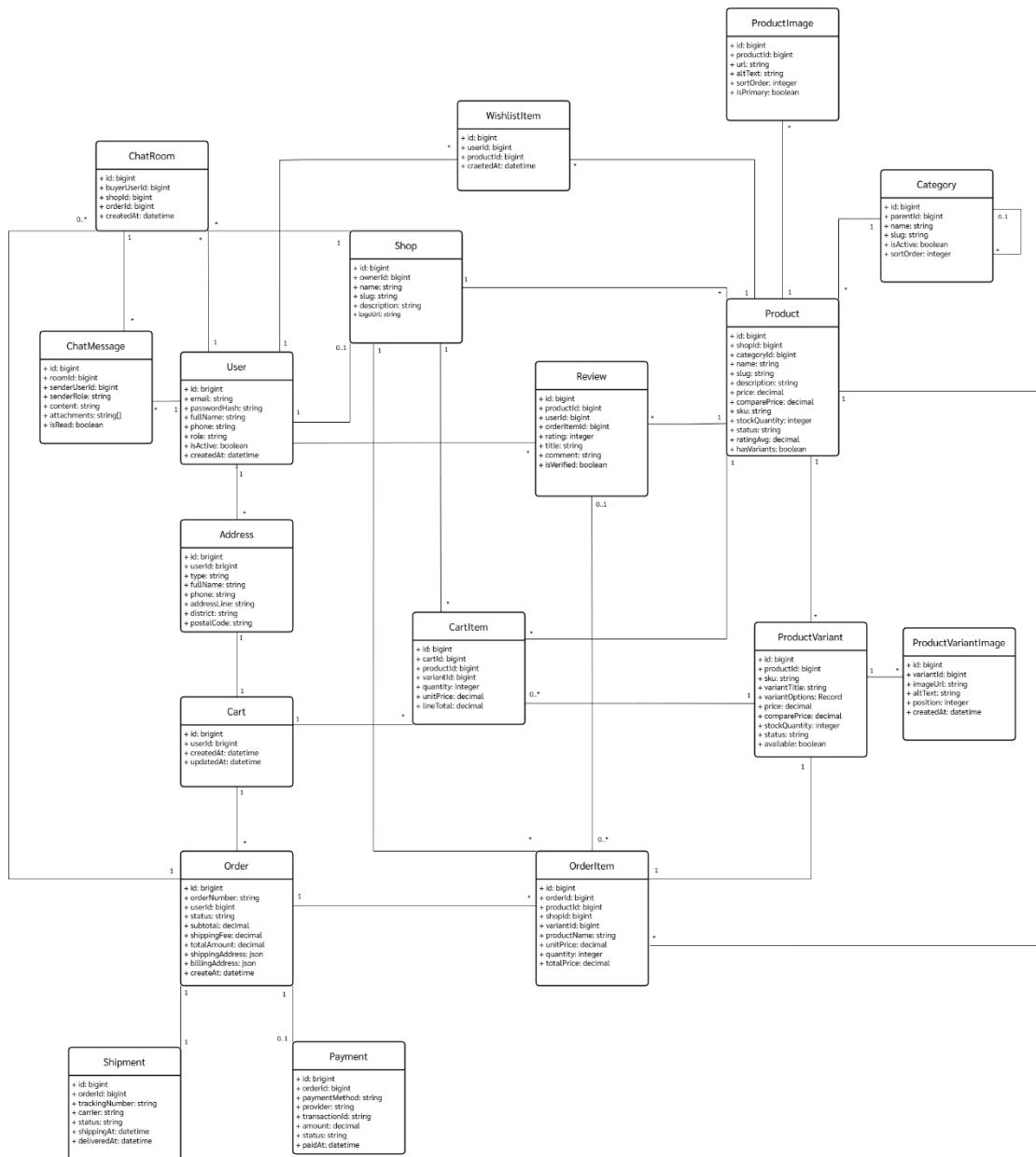


ภาพที่ 3 Usecase Diagram ของ User/Buyer



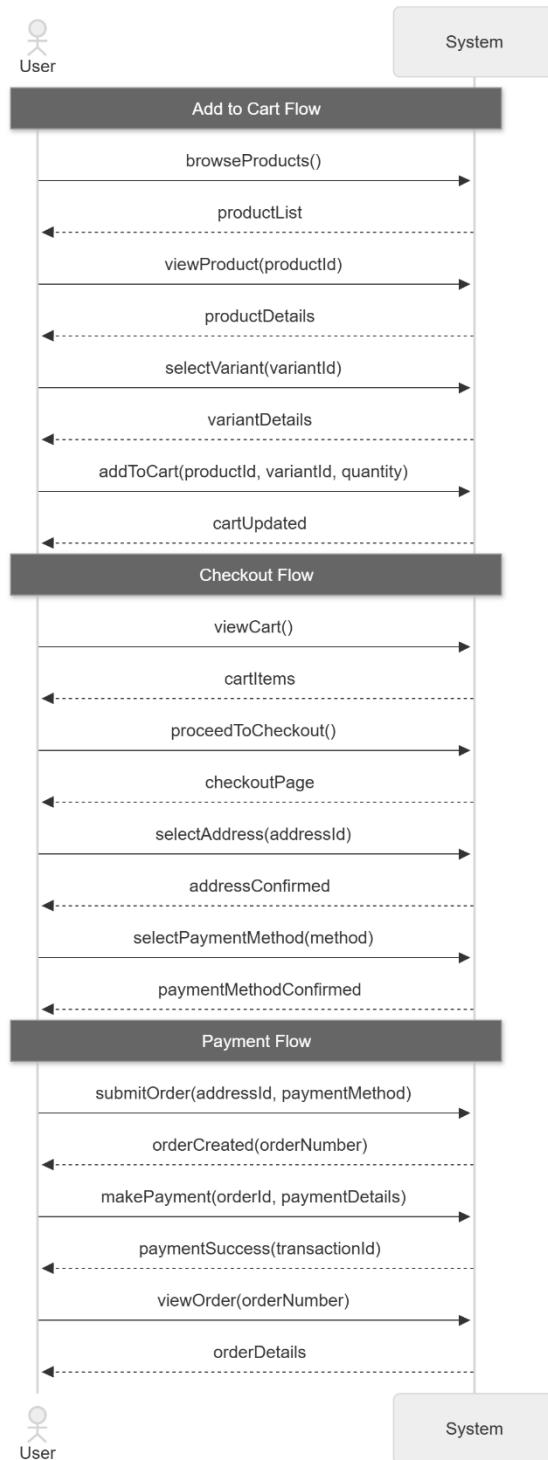
3.3.2 Domain Model

ภาพที่ 4 Domain Model

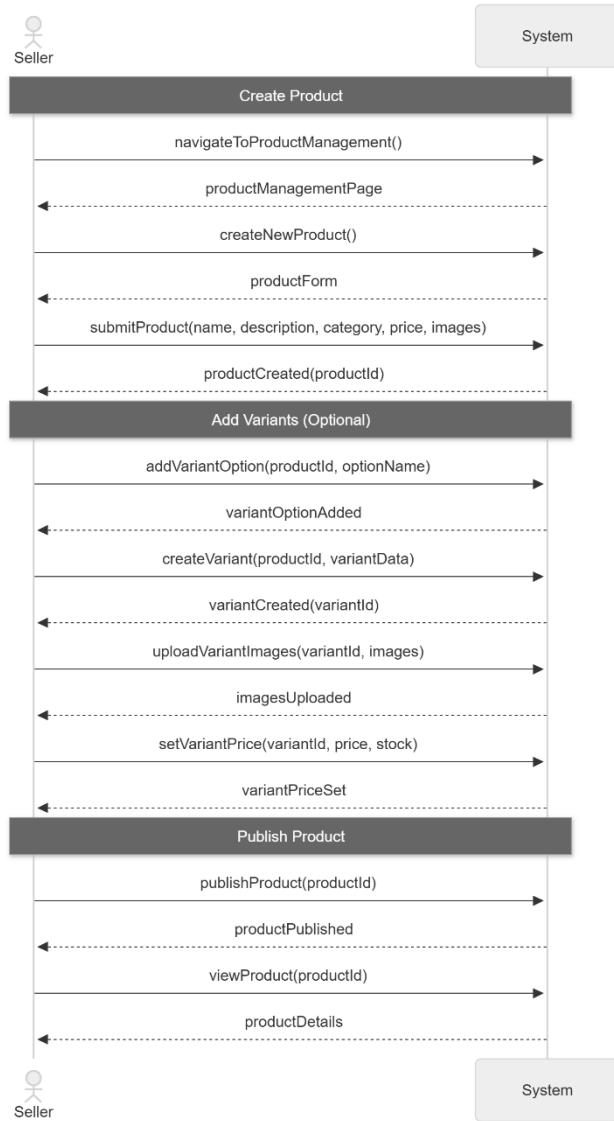


3.3.3 System Sequence diagram

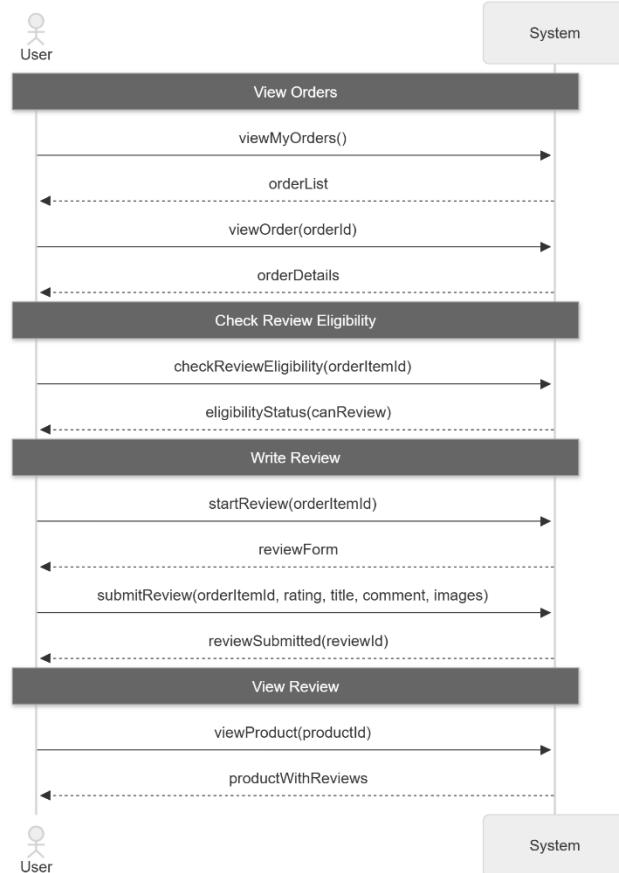
ภาพที่ 5 System Sequence diagram (User ซื้อสินค้า)



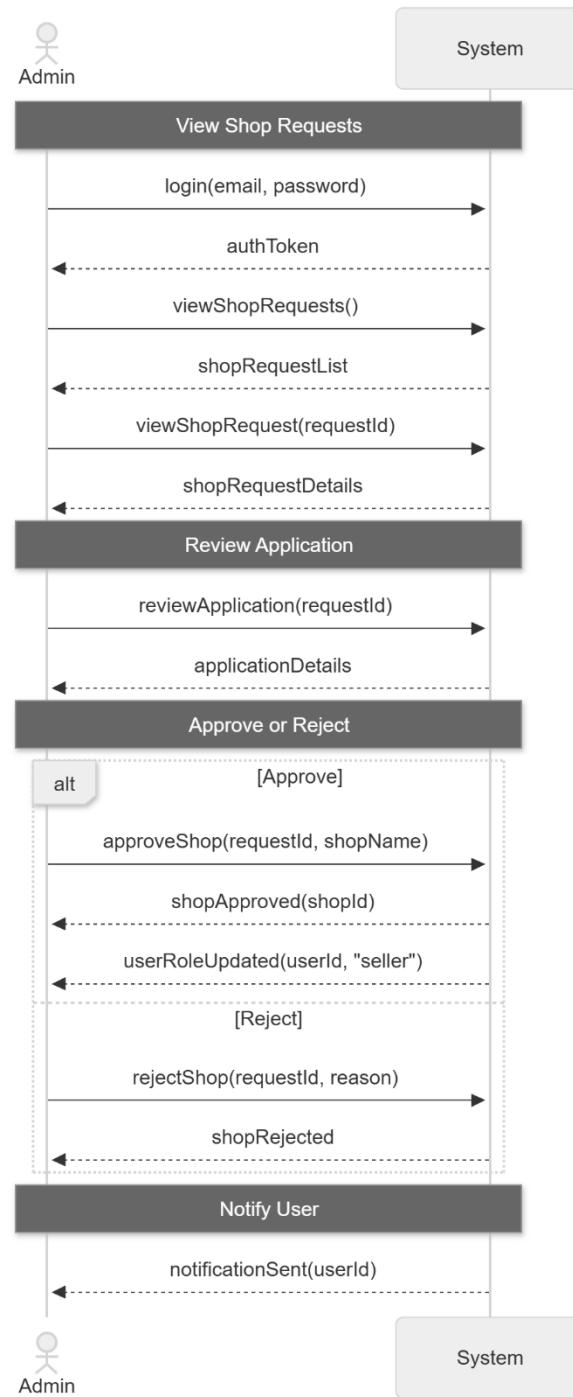
ภาพที่ 6 System Sequence diagram (Seller เพิ่มสินค้า)



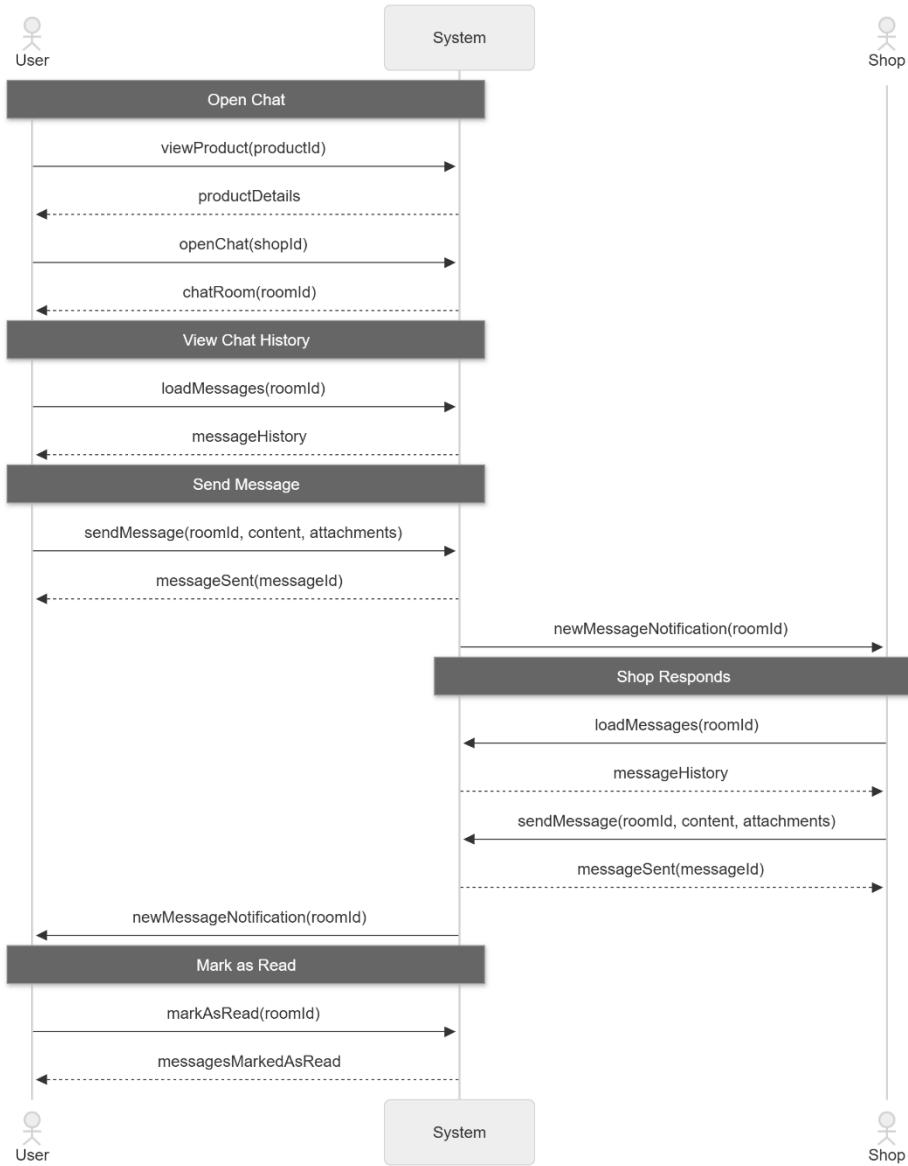
ภาพที่ 7 System Sequence diagram (User รีวิวสินค้า)



ภาพที่ 8 System Sequence diagram (Admin อนุมัติร้านค้า)

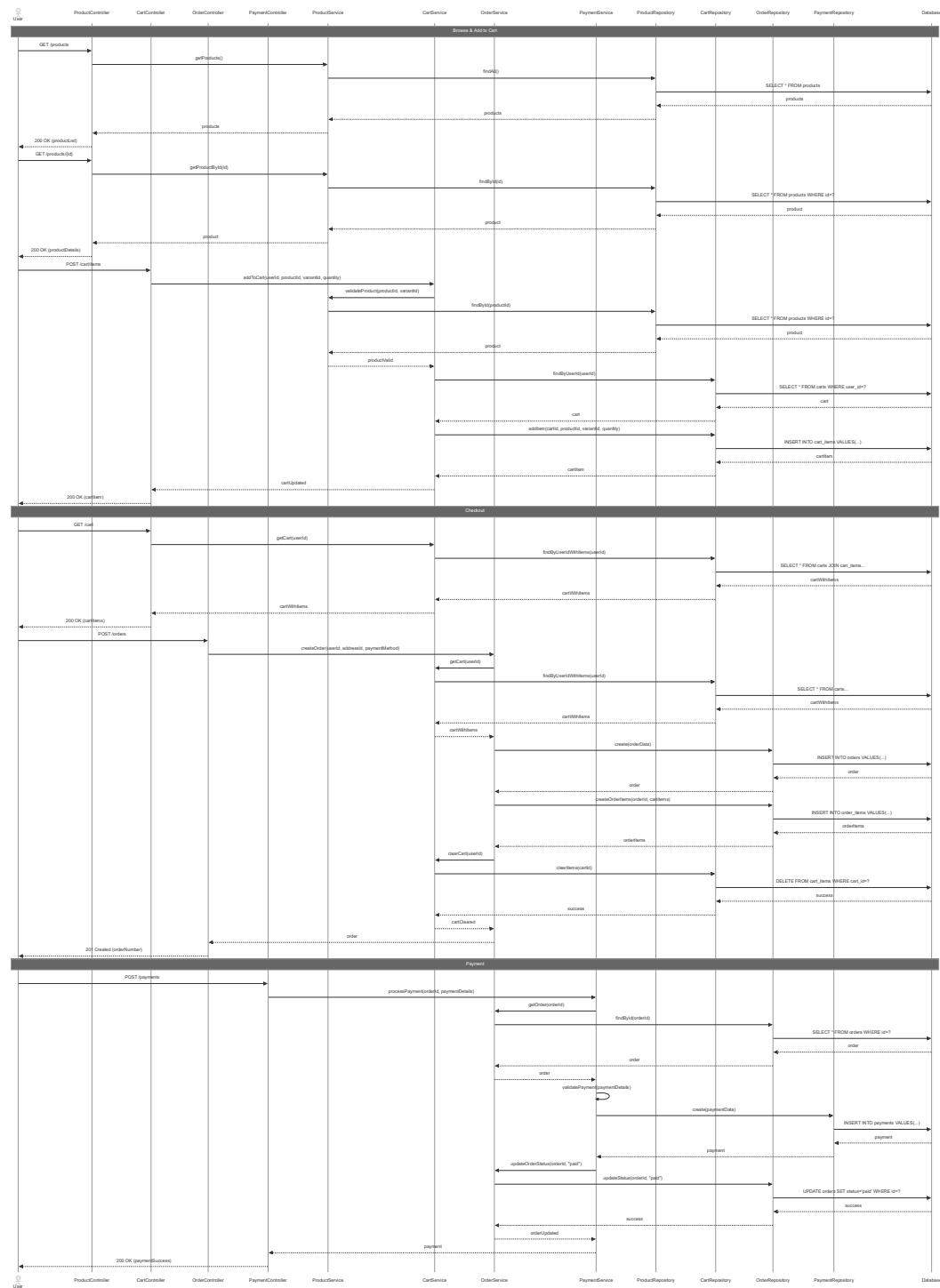


ภาพที่ 9 System Sequence diagram (User แลกเปลี่ยน Shop)

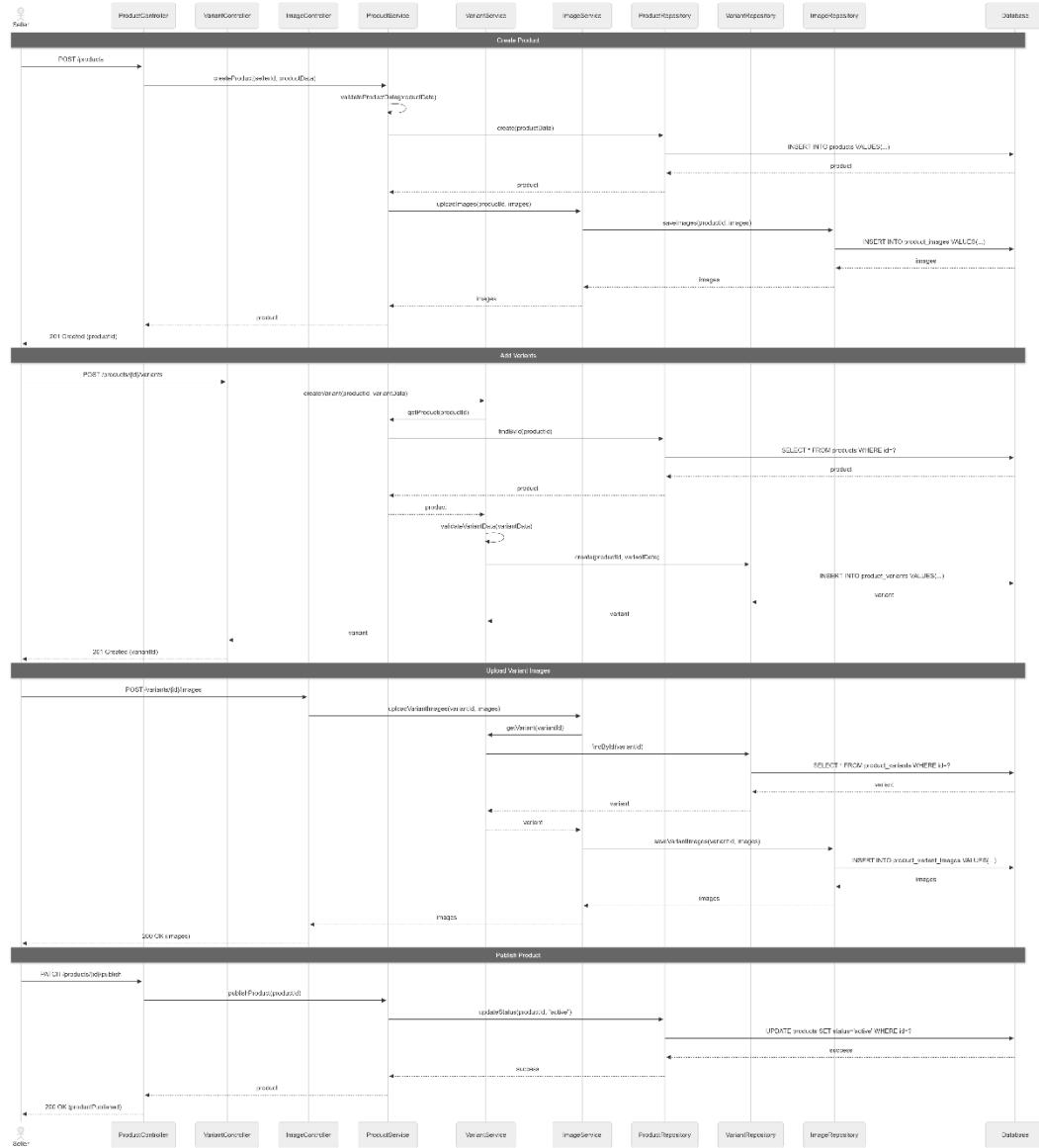


3.3.4 Sequence Diagram

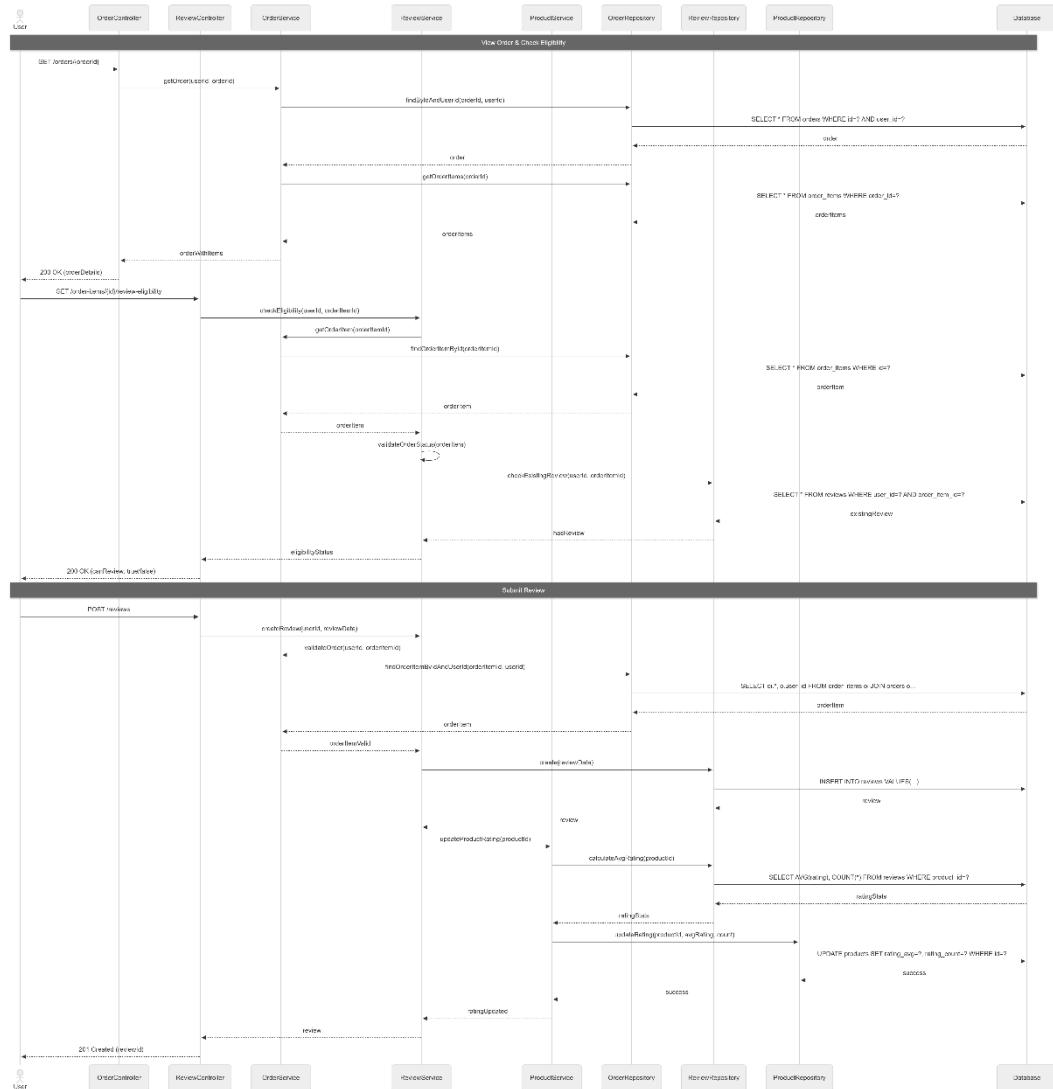
ภาพที่ 10 Sequence Diagram (User ซื้อสินค้า)



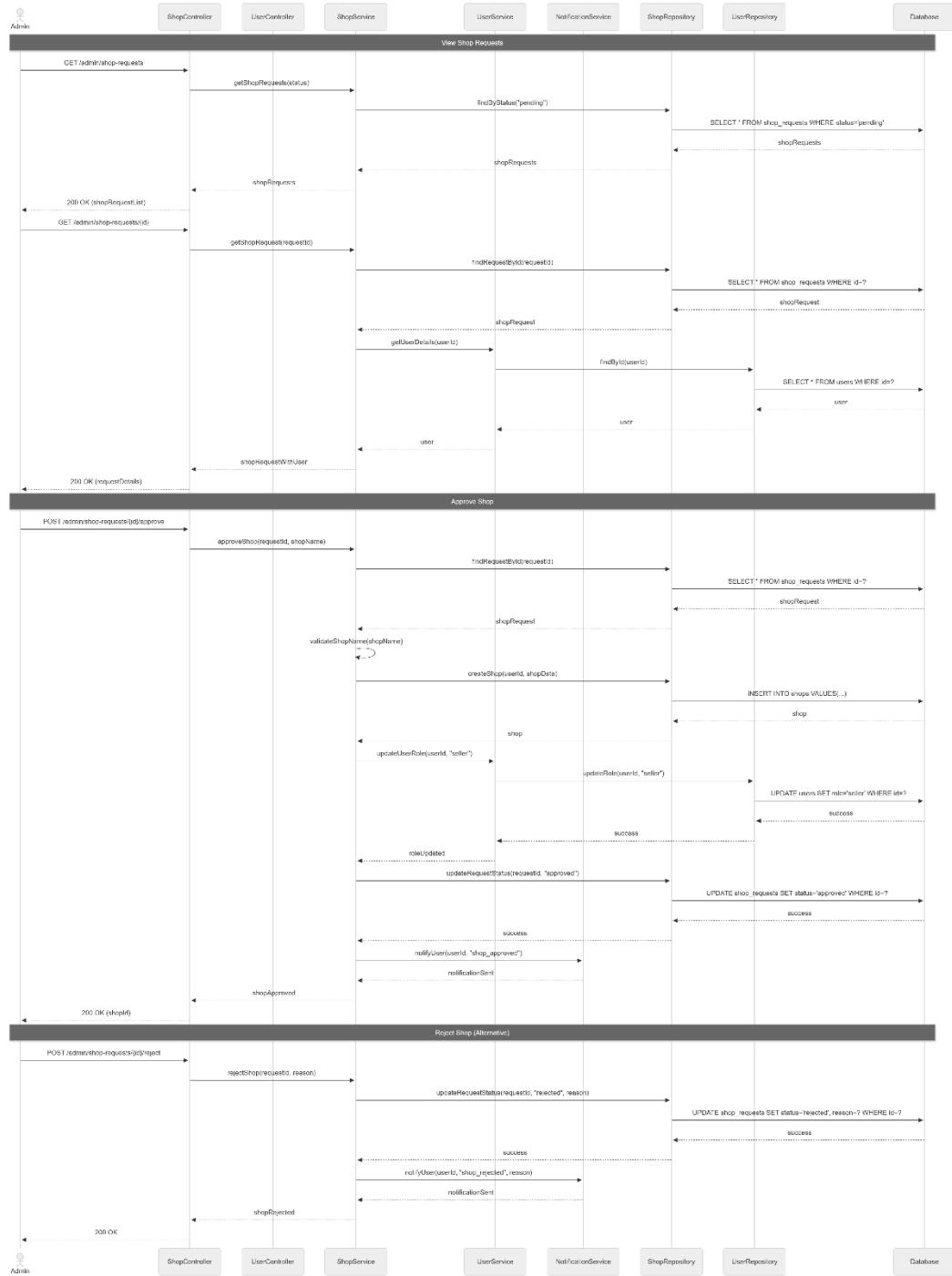
ภาพที่ 11 Sequence Diagram (Seller เพิ่มสินค้า)



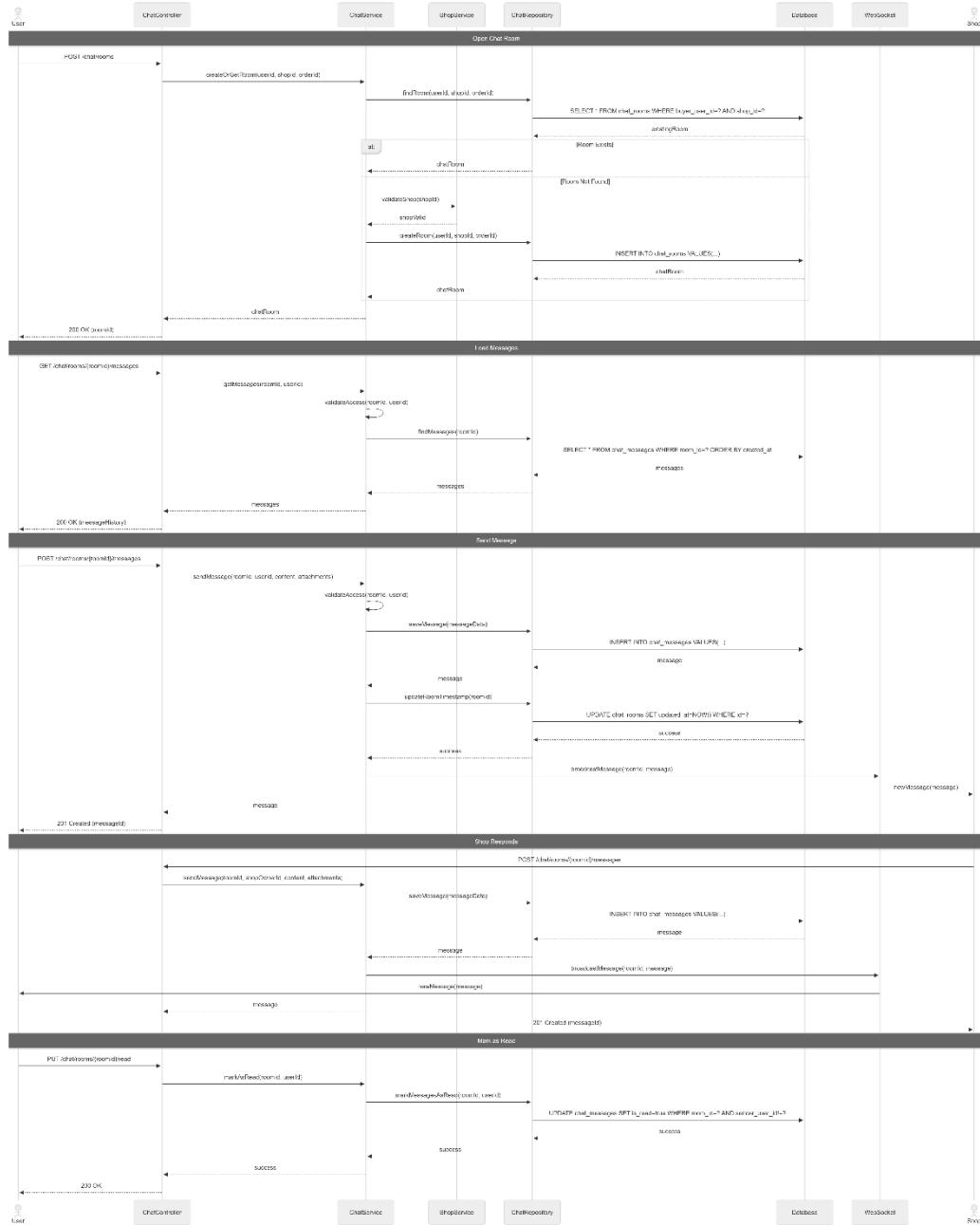
ภาพที่ 12 Sequence Diagram (User รีวิวสินค้า)



ภาพที่ 13 Sequence Diagram (Admin อนุมัติร้านค้า)

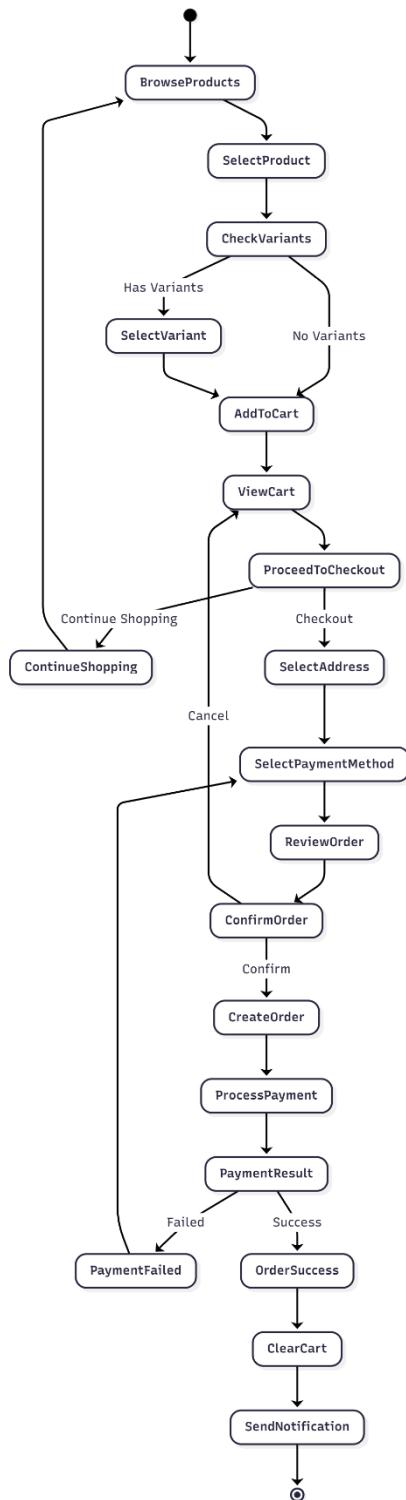


ภาพที่ 14 Sequence Diagram (User แชทกับ Shop)

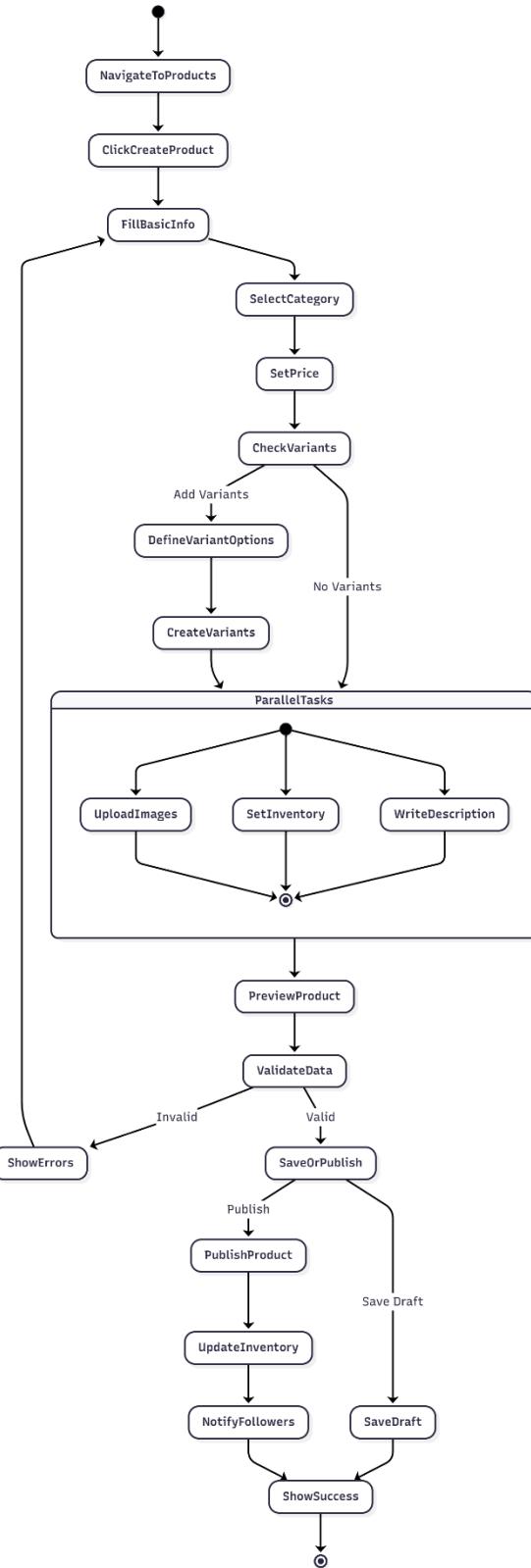


3.3.5 Activity Diagram

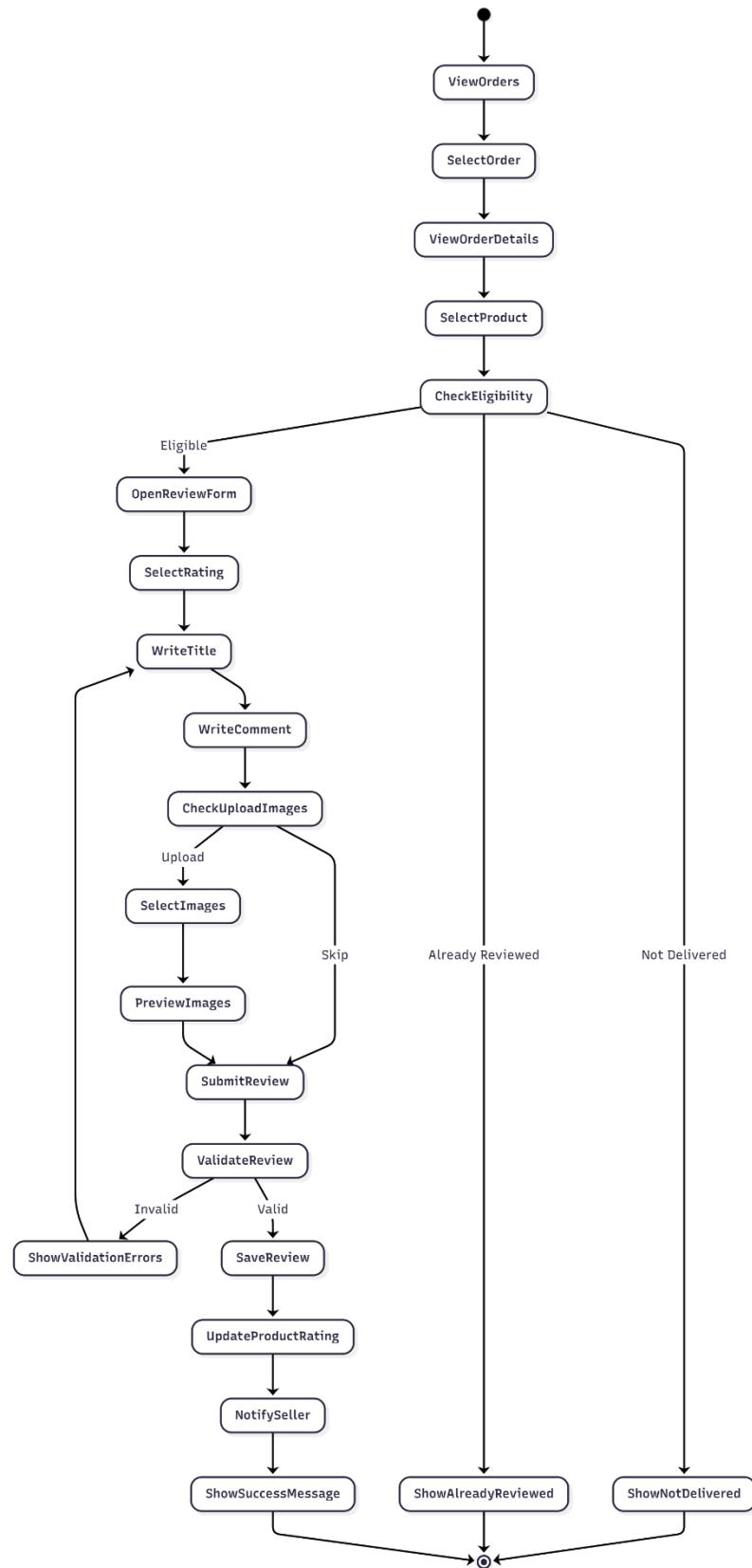
ภาพที่ 15 Activity Diagram (User ซื้อสินค้า)



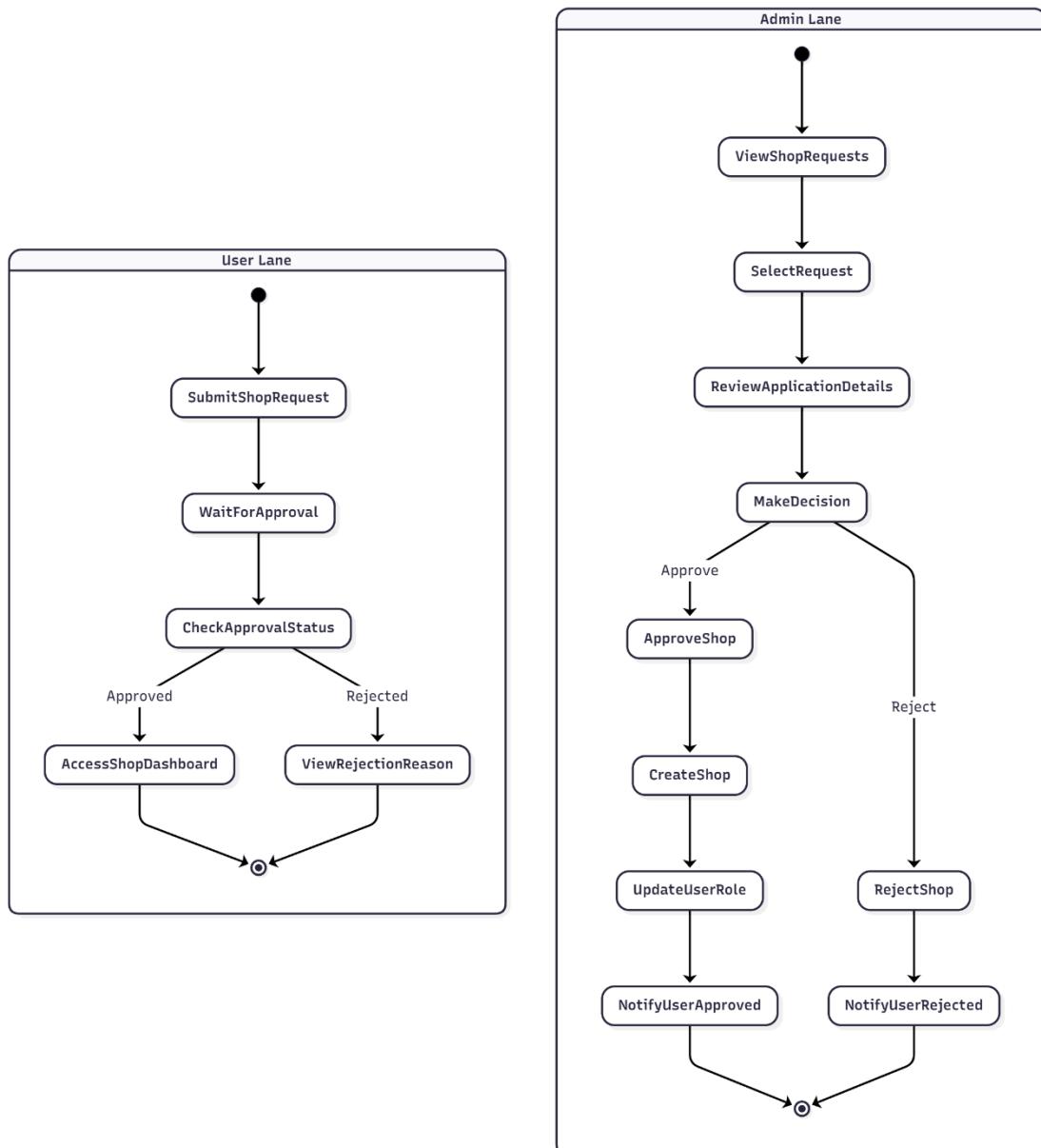
ภาพที่ 16 Activity Diagram (Seller เพิ่มสินค้า)



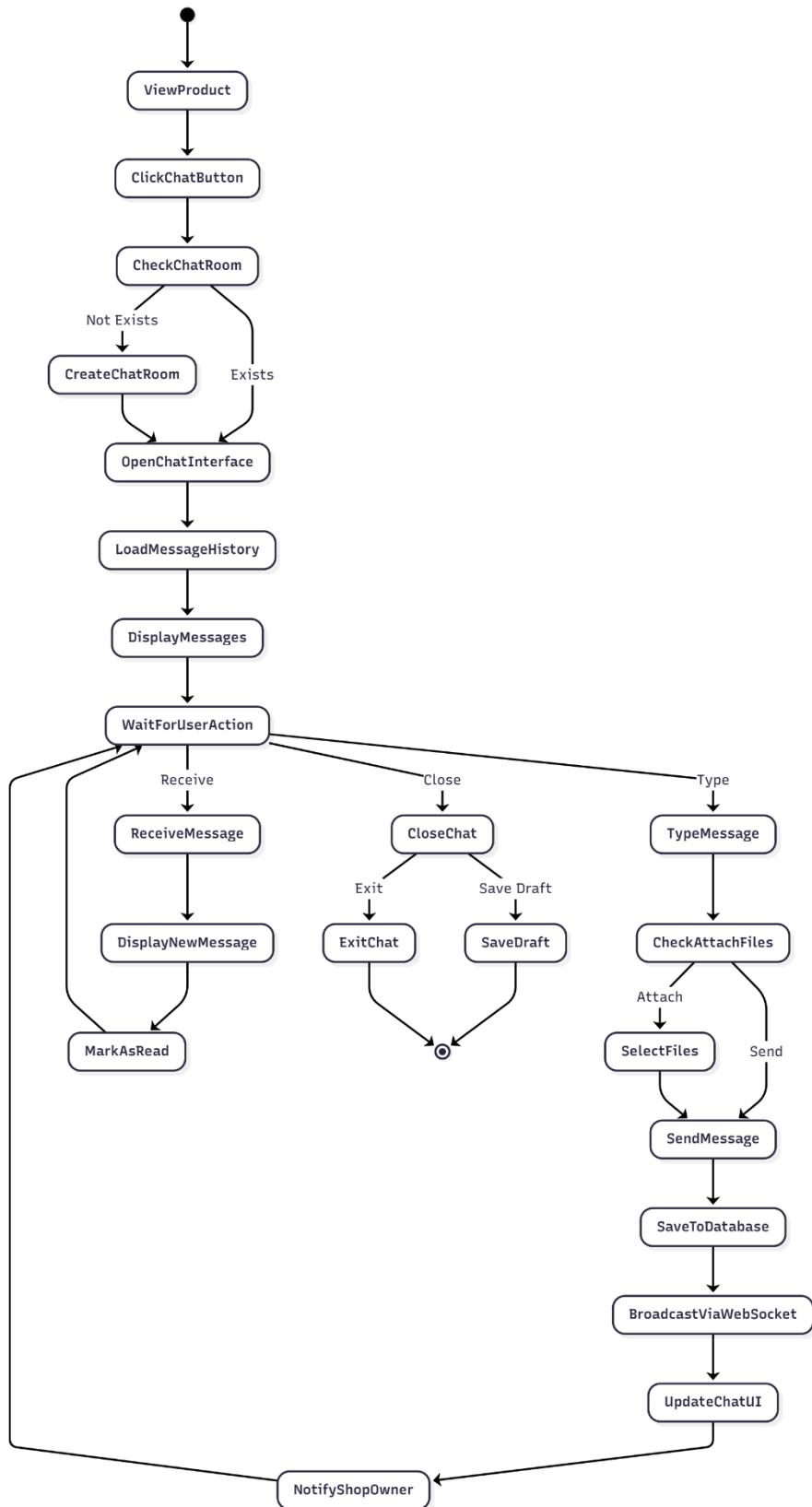
ภาพที่ 17 Activity Diagram (User รีวิวสินค้า)



ภาพที่ 18 Activity Diagram (Admin อนุมัติร้านค้า)



ภาพที่ 19 Activity Diagram (User แชทกับ Shop)

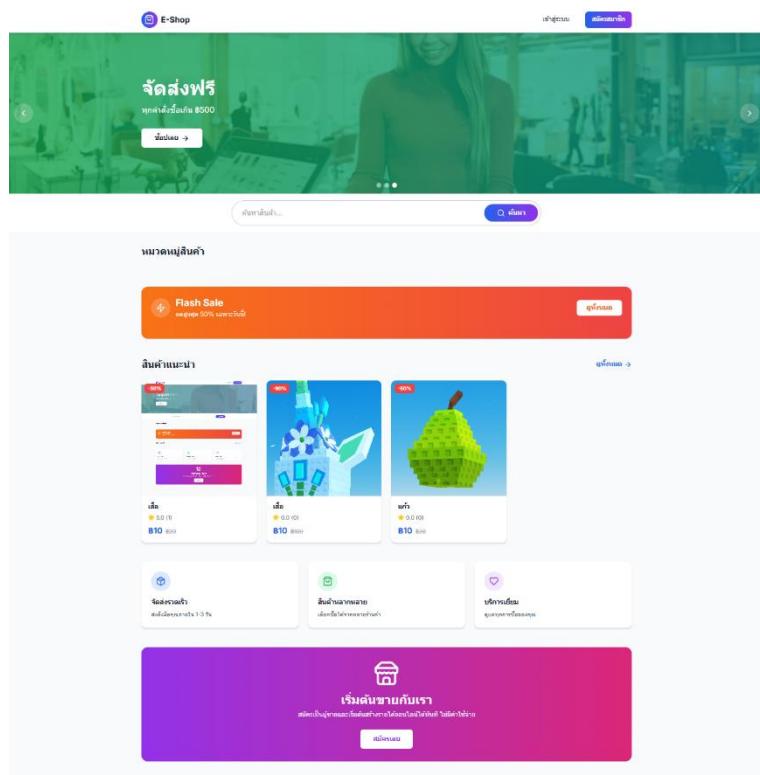


บทที่ 4

ผลการพัฒนาระบบ

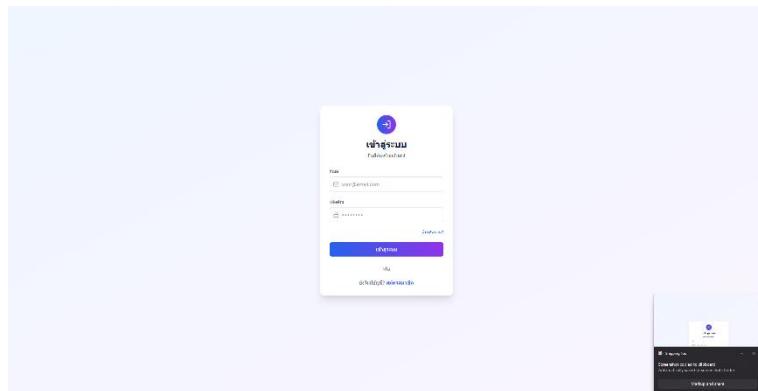
จากการดำเนินการวิเคราะห์และออกแบบแพลตฟอร์มสำหรับการซื้อขายผ่านช่องทางออนไลน์ ได้ทำการพัฒนาเป็นระบบที่สมบูรณ์ โดยมีการแบ่งสิทธิ์การใช้งานออกเป็น 3 ส่วน และมีการทำงานดังนี้

4.1 ส่วนของผู้ใช้งานทั่วไป (User)

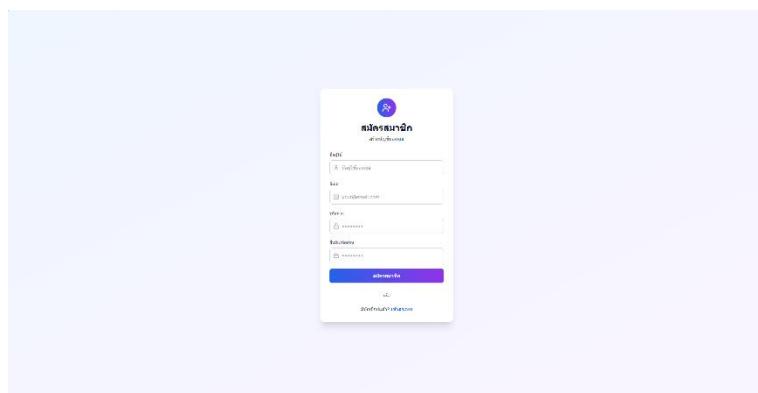


ภาพที่ 20 หน้าแรก

หน้าแรกของระบบเป็นหน้าที่ผู้ใช้งานเข้าสู่ได้ทันทีเมื่อเปิดเว็บไซต์ โดยแสดงข้อมูลสินค้าแนะนำ สินค้ายอดนิยม และโปรโมชั่นต่างๆ ผู้ใช้งานสามารถค้นหาสินค้าผ่านช่องค้นหา เลือกหมวดหมุสินค้า และเข้าสู่ระบบหรือสมัครสมาชิกได้โดย

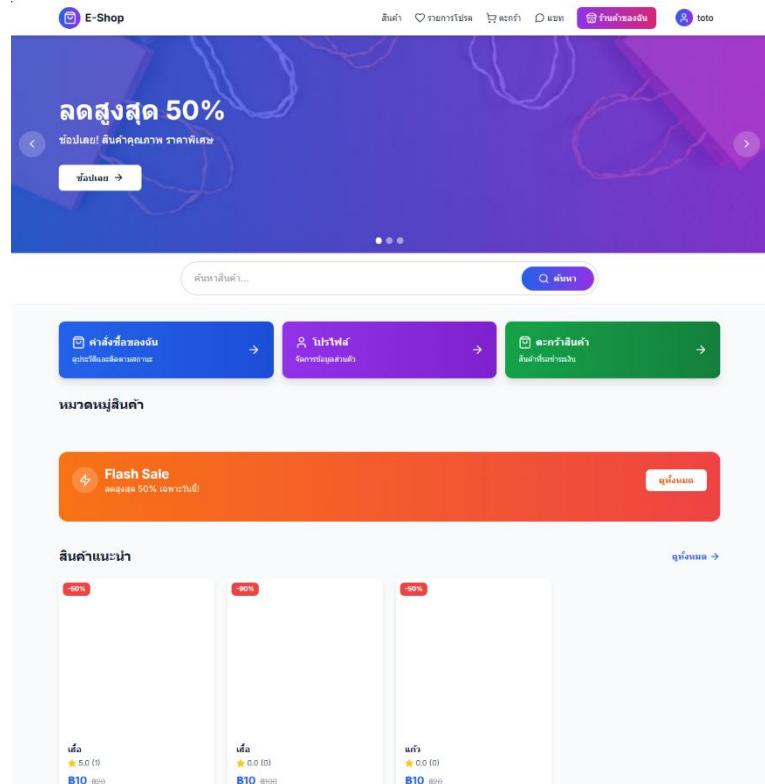


ກາພທີ 21 ຜັນຍາສັນຕິພາບສາມາຊີກແລະເຂົ້າສູ່ຮະບບ

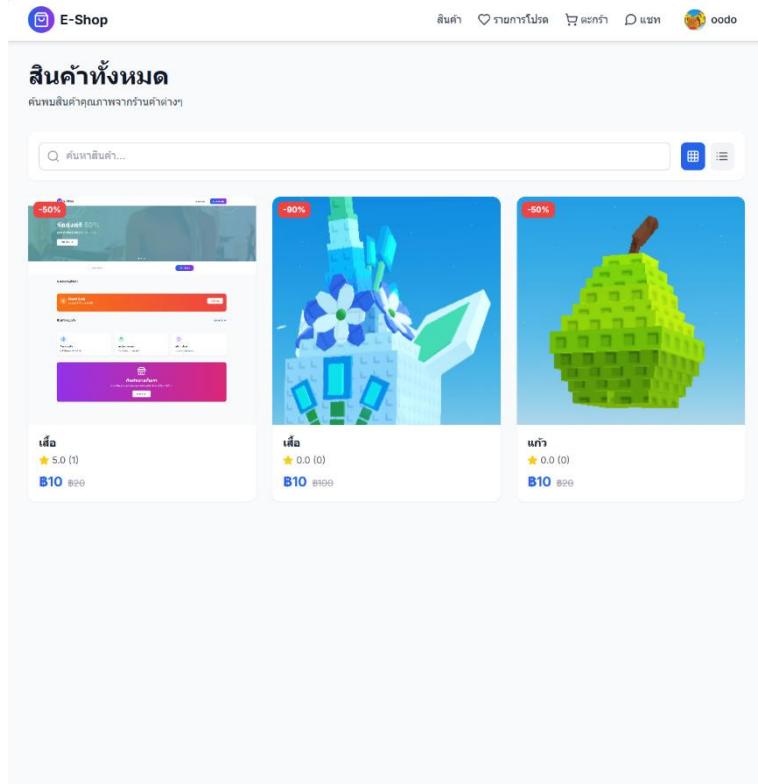


ກາພທີ 22 ຜັນຍາສັນຕິພາບສາມາຊີກແລະເຂົ້າສູ່ຮະບບ (ຕ່ອ)

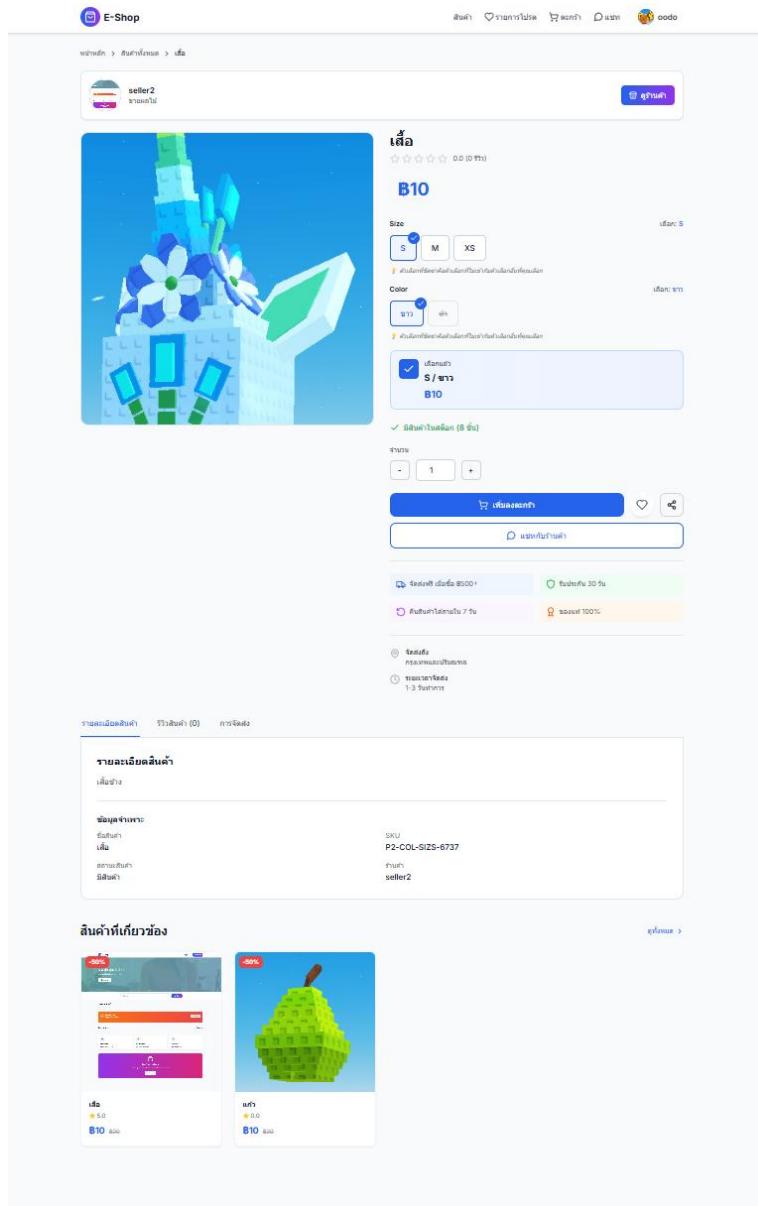
ໜ້າສໍາຫຼັບຜູ້ໃໝ່ໃນມໍ່ສັນຕິພາບສາມາຊີກແລະຜູ້ໃໝ່ໃໝ່ເດີມເຂົ້າສູ່ຮະບບ ໃນການສັນຕິພາບສາມາຊີກຕ້ອງທ່ານກອກ
ຂໍ້ມູນລ່ວມຕົວ ແລະຄລິກປຸ່ມ "ສັນຕິພາບສາມາຊີກ" ຮະບບະຈະນຳໄປໜ້າເຂົ້າສູ່ຮະບບ ແລະສໍາຫຼັບການເຂົ້າສູ່ຮະບບຕ້ອງທ່ານກອກ
ກອກອື່ນເລີລແລະຮ້ສັ່ນ ຈາກນັ້ນຄລິກປຸ່ມ "ເຂົ້າສູ່ຮະບບ"



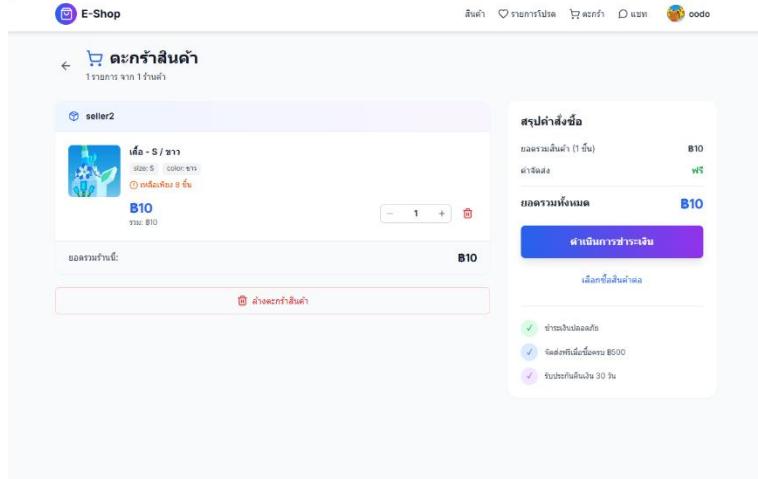
ภาพที่ 23 หน้าหลักจากการเข้าสู่ระบบ เมื่อทำการเข้าสู่ระบบจะพบกับหน้านี้



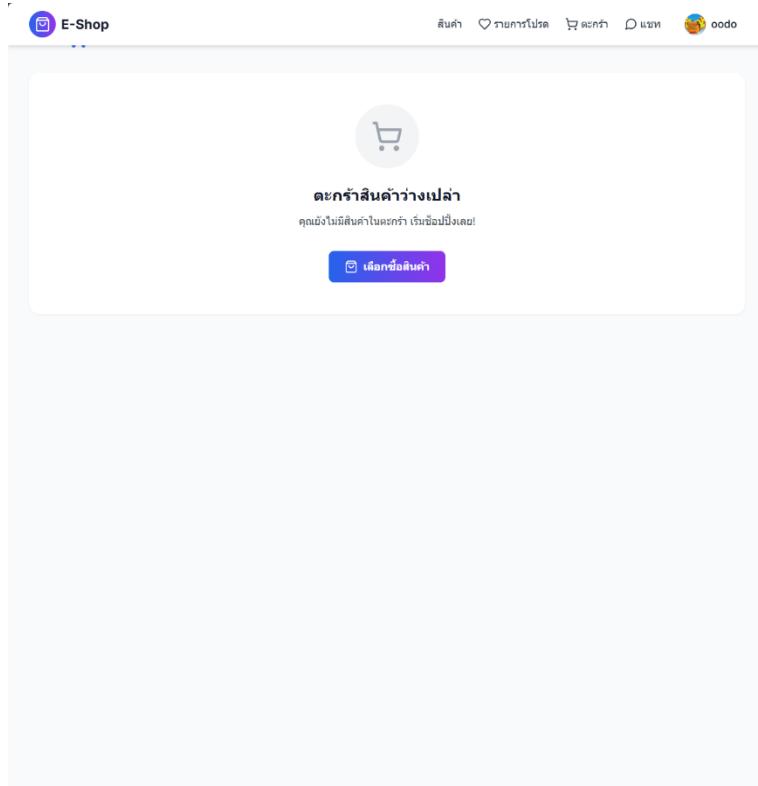
ภาพที่ 24 หน้ารายการสินค้า โดยเป็นหน้าแสดงรายการสินค้าทั้งหมด มีระบบค้นหา กรองและเรียงลำดับสินค้า



ภาพที่ 25 หน้ารายละเอียดสินค้า เป็นหน้าแสดงข้อมูลสินค้าแบบละเอียด รวมถึงภาพสินค้า ราคา คำอธิบาย และรีวิวจากผู้ซื้อ



ภาพที่ 26 หน้าต่างกร้าสินค้า เป็นหน้าแสดงรายการสินค้าที่ผู้ใช้งานเลือกไว้ พร้อมคำนวณราคารวม



ภาพที่ 27 หน้าต่างกร้าสินค้า (ต่อ) ตัวอย่างในกรณีที่ไม่มีสິນຄ້າໃນທະກຳ

E-Shop
สินค้า
รายการโปรด
ออกจากร้าน
แบบ
 0000

← กดเพื่อไปที่หน้าจอเดิม

ข้ารະເປັນ

1 รายการໃນສະຕິພາບ



ສິນຄ້າທີ່ຢູ່ຂອງຂ່າຍ



ຊ່າງລວມ



ສະວັດລວມເຈົ້າຂໍ້ຕົ້ນ

 **ສິນຄ້າທີ່ຢູ່ຂອງຂ່າຍ**

 ເລືອກໂທໃຫຍ່ທີ່ບໍ່ມີໃຫຍ່

ສິນຄ້າທີ່ຢູ່ຂອງຂ່າຍ

ເລີດຕາມເສີມຜ່ານ (1 ຕົ້ນ) **B10**

ຜ່ານສິນສະໜັບ **B50**

👉 ສິນຄ້າ B400 ເລີດຕາມຜ່ານ

ມອບຄວາມສິ່ງມາດ **B60**

 ລ່າຍວັນໄສລະຍຸດກົດ

 ຮັບສິນທີ່ເຈົ້າລວມ B500

 ອີເປີຍກົດເປັນໃນ 30 ວັນ

ภาพที่ 28 หน้าชำระเงิน ในส่วนแรกเป็นหน้าสำหรับกรอกข้อมูลการจัดส่ง

← กดสีไปรษณีย์
ชื่าระเงิน
รายการใบเรียกเข้า

ที่อยู่บ้านเด็กน้ำ

ชื่อบ้านเด็กน้ำ	เบอร์โทรศัพท์	อีเมล	สถานะ
555 สำหรับ 5555555555	555	indy@example.com	B10
5555555555 55 55	55555555		B50

สถานะสั่งซื้อ

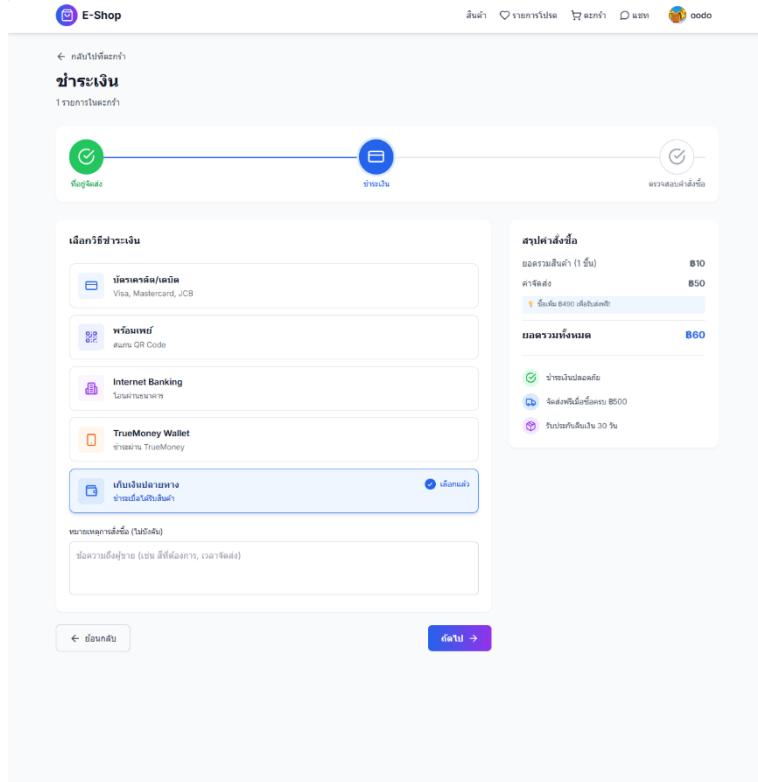
ยอดรวมเดือน (1 เดือน)	จำนวนเดือน
จำนวนเงิน ฿400 จำนวนเดือน 1	จำนวนเดือน 1

ผลรวมเงินหมด

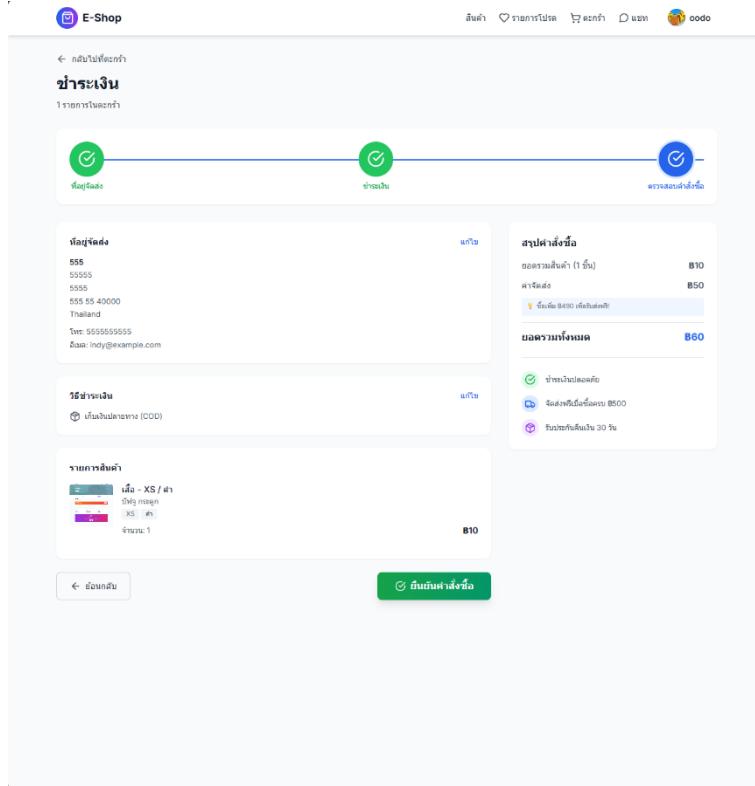
จำนวนเดือน	จำนวนเดือน	จำนวนเดือน
จำนวนเดือน 1	จำนวนเดือน 1	จำนวนเดือน 30 วัน

ต่อไป >

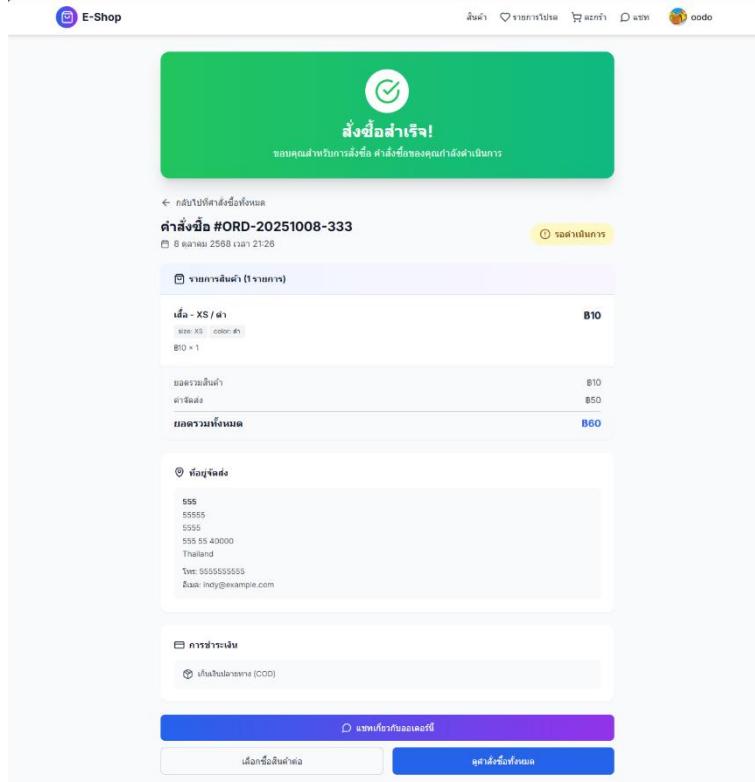
ภาพที่ 29 หน้าชื่าระเงิน (ต่อ)



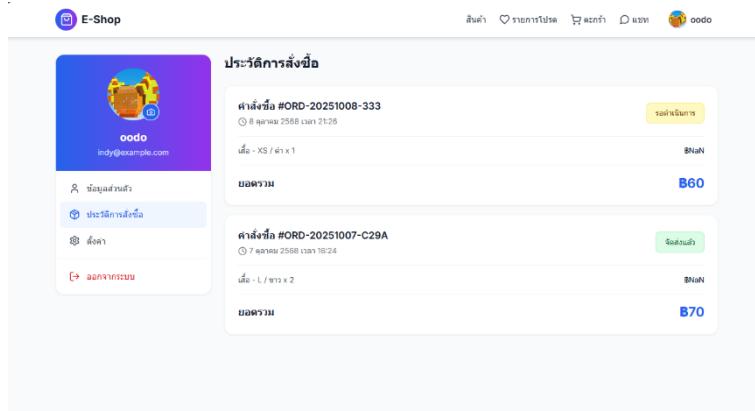
ภาพที่ 30 หน้าชำระเงิน (ต่อ) ในส่วนที่ 2 จะเป็นส่วนของการเลือกวิธีการชำระเงิน



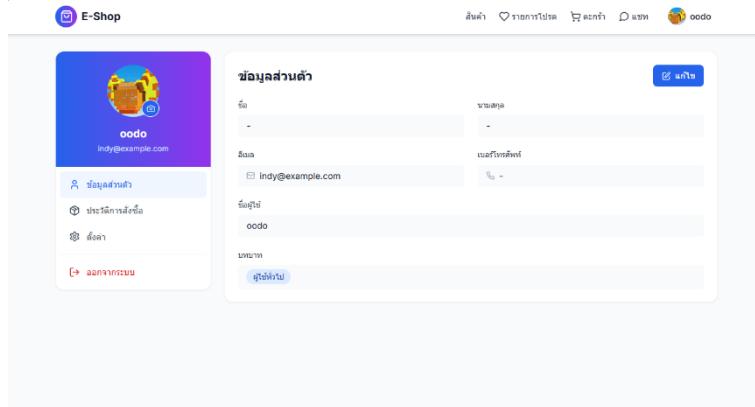
ภาพที่ 31 หน้าชาร์มเงิน (ต่อ) ในส่วนสุดท้าย จะเป็นส่วนของการตรวจสอบรายการสั่งซื้อ



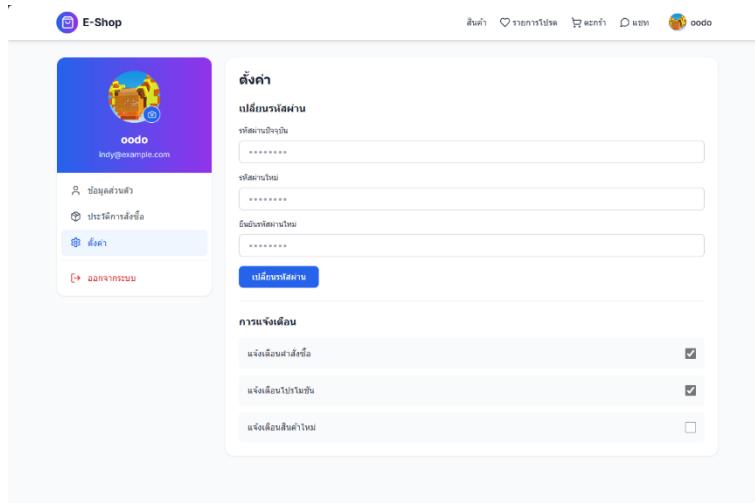
ภาพที่ 32 หน้าชำระเงิน (ต่อ) เมื่อยืนยันการสั่งซื้อ จะปรากฏหน้าแสดงข้อความ “สั่งซื้อสำเร็จ” และรายละเอียดการสั่งซื้อ



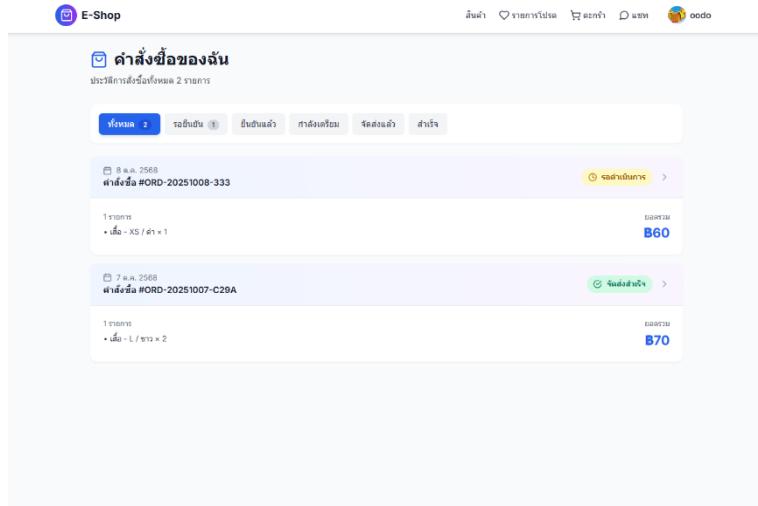
ภาพที่ 33 หน้าประวัติการสั่งซื้อ หน้าแสดงรายการคำสั่งซื้อทั้งหมดของผู้ใช้งาน โดยเข้าได้จากส่วนของໂປຣໄຟລ໌



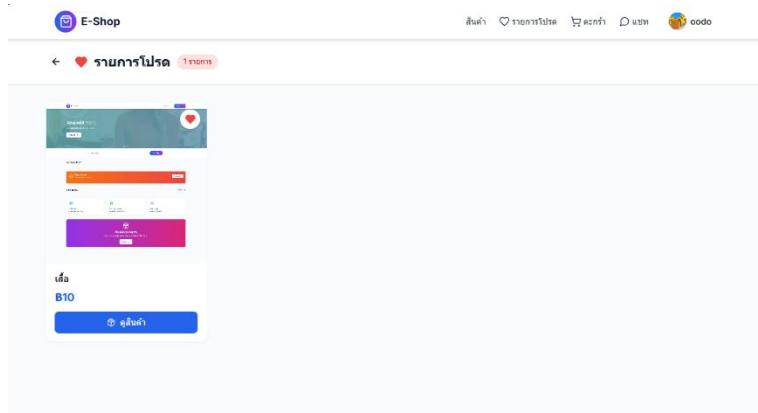
ภาพที่ 34 หน้าข้อมูลส่วนตัว โดยเข้าดูได้จากส่วนของโปรไฟล์



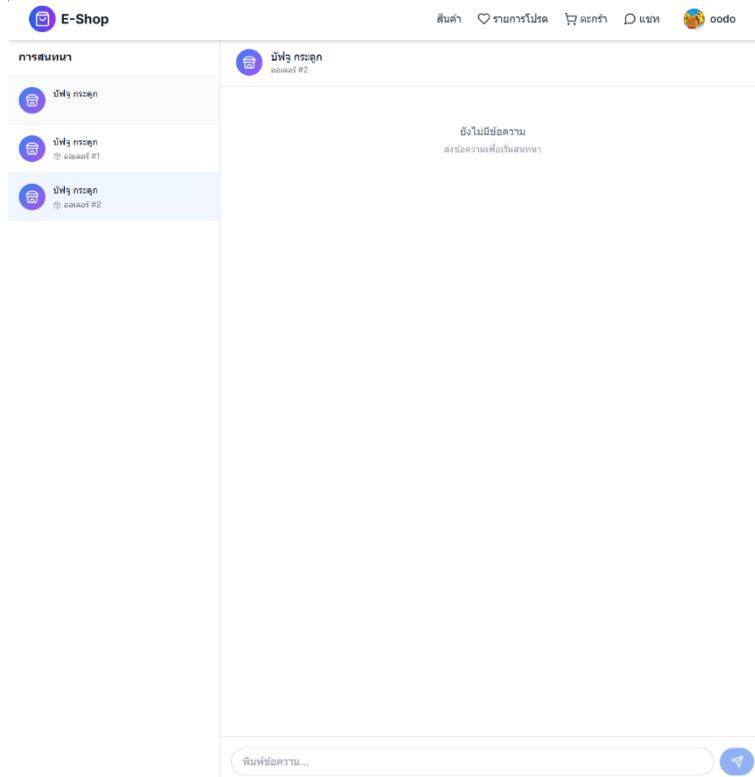
ภาพที่ 35 หน้าตั้งค่า โดยเข้าดูได้จากส่วนของโปรไฟล์



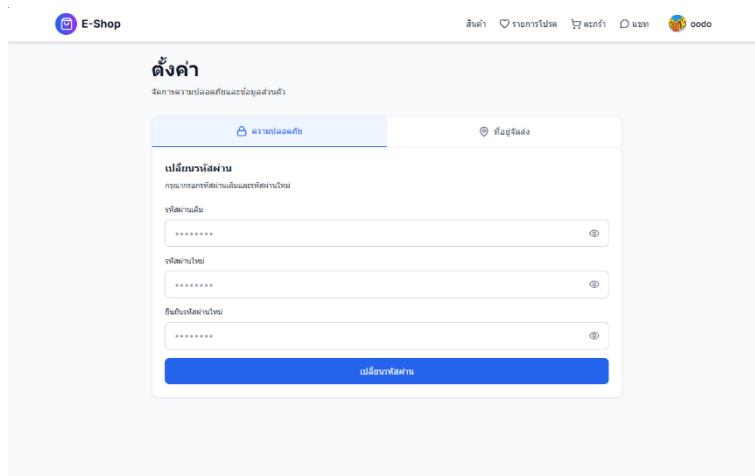
ภาพที่ 36 หน้าคำสั่งซื้อของฉัน สำหรับการเข้าดูคำสั่งซื้อที่เกิดขึ้น จัดการรายการสั่งซื้อ



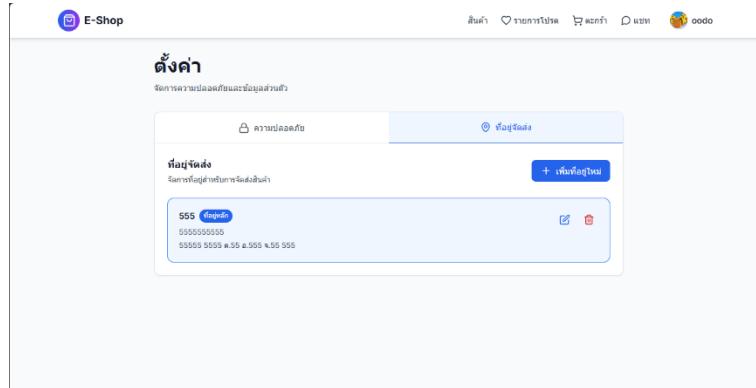
ภาพที่ 37 หน้ารายการโปรด เป็นหน้าที่แสดงรายการสินค้าที่ผู้ใช้ทำการกดถูกใจได้



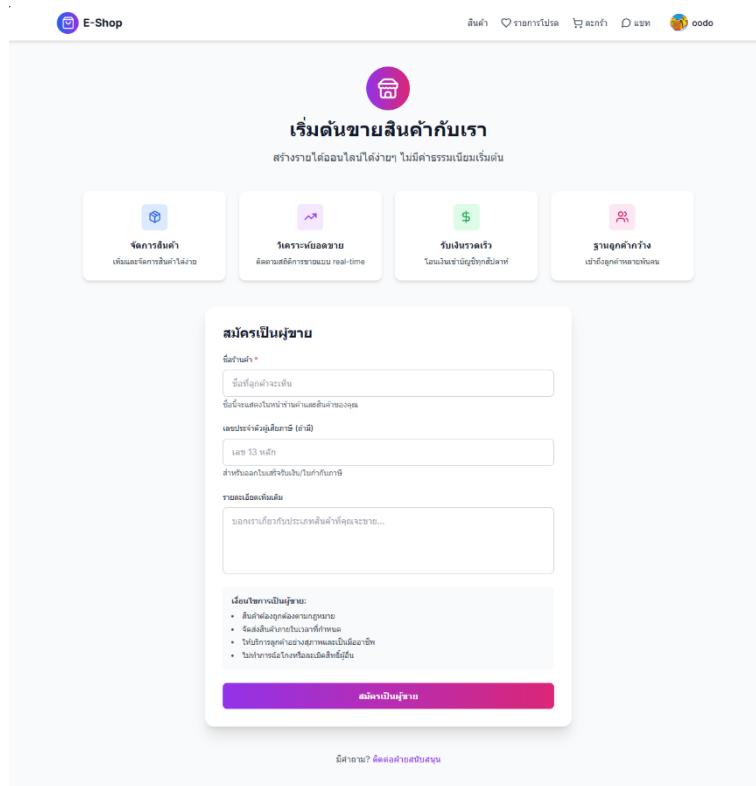
ภาพที่ 38 หน้าแดชบอร์ดของผู้ใช้ เป็นหน้าสำหรับการติดต่อ กับผู้ขาย



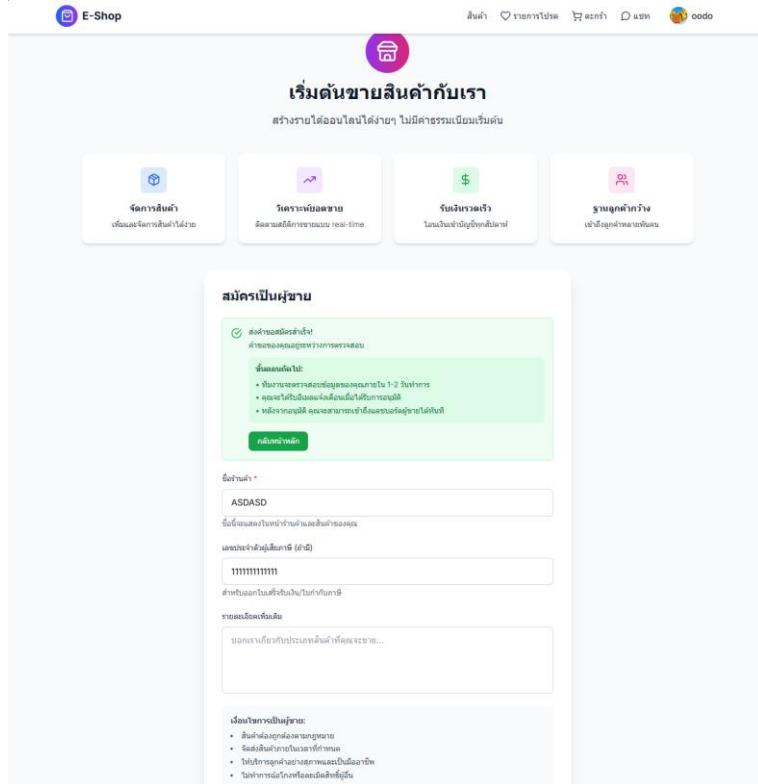
ภาพที่ 39 หน้าตั้งค่า ในส่วนแรกเป็นการตั้งค่ารหัสผ่าน



ภาพที่ 40 หน้าตั้งค่า (ต่อ) ในส่วนที่สองจะเป็นการตั้งค่าที่อยู่ในการจัดส่ง

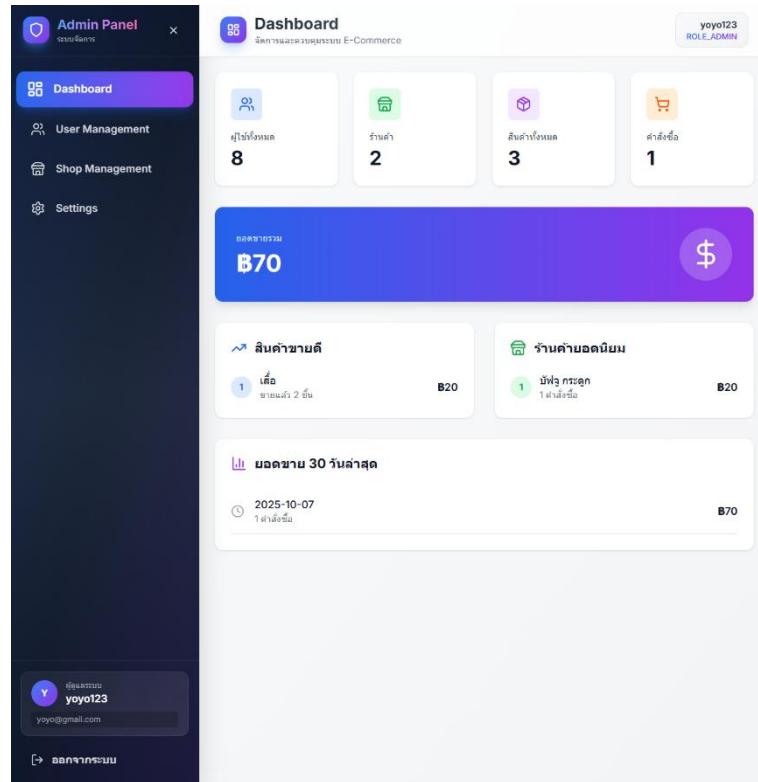


ภาพที่ 41 หน้าเปิดร้านค้า ผู้ใช้สามารถสมัครเป็นผู้ขาย โดยกรอกรายละเอียด ได้แก่ ชื่อร้านค้า เลขประจำตัวผู้เสียภาษี และรายละเอียดเพิ่มเติม และกด “สมัครเป็นผู้ขาย”



ภาพที่ 42 หน้าเปิดร้านค้า (ต่อ) เมื่อกด “สมัครเป็นผู้ขาย” จะเป็นการขอการอนุมัติจากผู้ดูแลระบบ ในระหว่างนี้ ผู้ใช้จำเป็นต้องรอ เมื่อผู้ดูแลระบบยืนยัน จึงจะสามารถเปิดร้านค้าได้

4.2 ส่วนของผู้ดูแลระบบ (Admin)



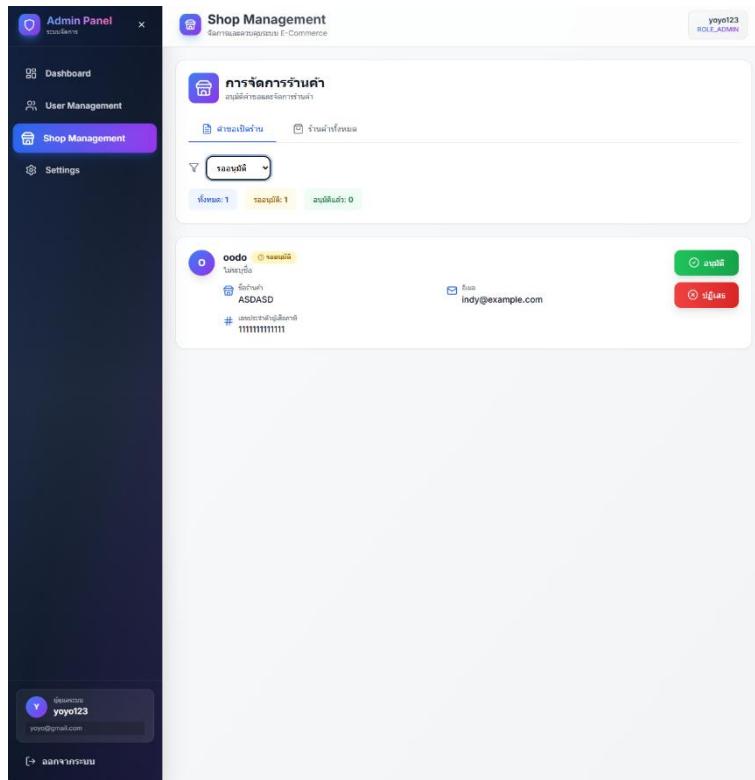
ภาพที่ 43 หน้าแดชบอร์ดของผู้ดูแลระบบ ในหน้านี้จะแสดงภาพรวมของระบบทั้งหมด

ชื่อ	อีเมล	Role	สถานะ	ตัด甭
suphakit No name	suphakit085@gmail.com	User	IsActive	7/10/2568
yoyo No name	thitikorn@example.com	User	IsActive	7/10/2568
Nik No name	nik@gmail.com	User	IsActive	7/10/2568
yoyo123 No name	yoyo@gmail.com	Admin	IsActive	7/10/2568
toto No name	toto@gmail.com	Seller	IsActive	7/10/2568
NickyAmp No name	nickyamp@gmail.com	Seller	IsActive	7/10/2568
odo No name	indy@example.com	User	IsActive	7/10/2568
seller2 No name	seller2@gmail.com	User	IsActive	7/10/2568

ผู้ดูแลระบบ
yoyo123
yoyo@gmail.com

[↗ ออกจากระบบ]

ภาพที่ 44 หน้าการจัดการผู้ใช้ ในหน้านี้จะแสดงเกี่ยวกับข้อมูลผู้ใช้ทั้งหมดในระบบ



ภาพที่ 45 หน้าการจัดการร้านค้า เมื่อผู้ดูแลเข้าสู่หน้านี้ ในส่วนแรกจะปรากฏคำขอเปิดร้านค้าที่รอดำเนินการ



ภาพที่ 46 หน้าการจัดการร้านค้า (ต่อ) เมื่อผู้ดูแลระบบอนุมัติคำขอ สถานะของผู้ใช้จะเปลี่ยนจาก 'User' เป็น 'Seller' และจัดการเกี่ยวกับร้านค้าได้

The screenshot displays two side-by-side panels from an Admin Panel application.

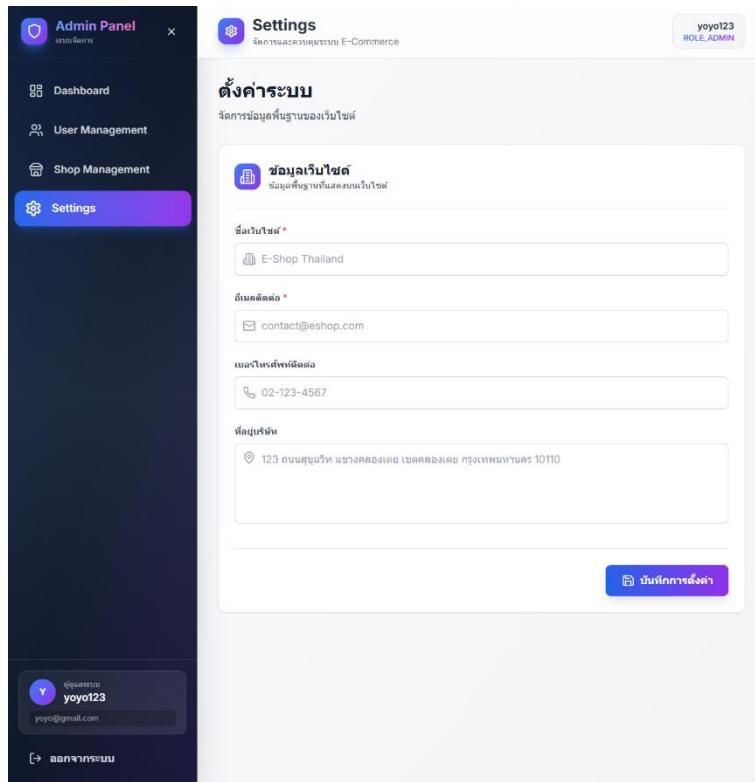
Left Panel (Admin Panel):

- Header: Admin Panel ภาษาไทย
- Navigation menu:
 - Dashboard
 - User Management
 - Shop Management** (highlighted in purple)
 - Settings
- User info: ผู้ดูแลระบบ yoyo123 ROLE_ADMIN
- Footer: ออกจากระบบ

Right Panel (Shop Management):

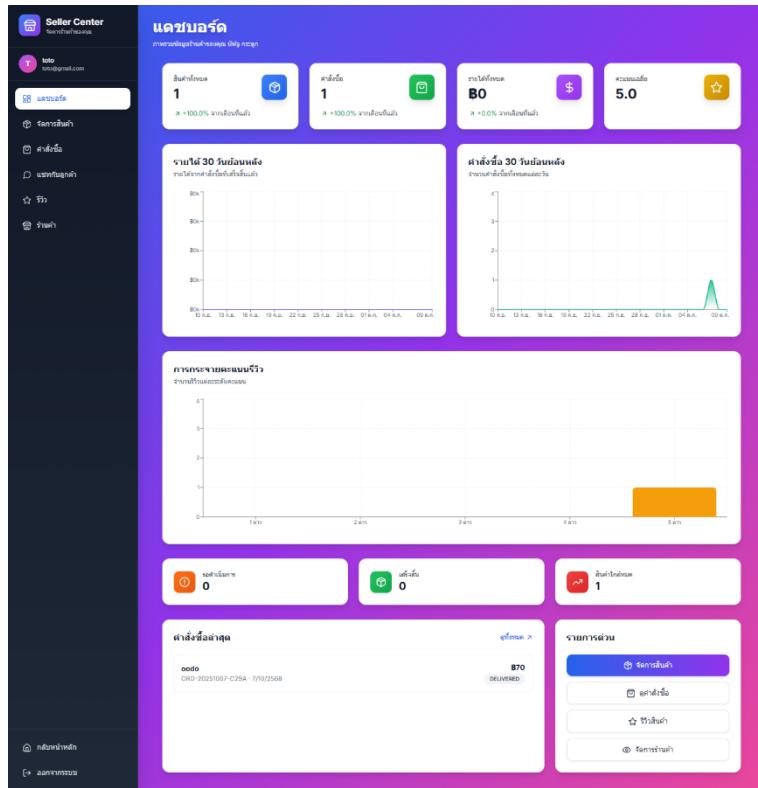
- Header: Shop Management จัดการร้านค้าของคุณ E-Commerce yoyo123 ROLE_ADMIN
- Section: การจัดการร้านค้า ของคุณ ล่าสุดและล่าช้า
- Sub-section: คำขอเปิดร้าน ร้านค้าทั้งหมด
- Filter: ทั้งหมด
- Count: ทั้งหมด: 3 เปิดรับ: 3 ยกเลิก: 0
- List of shop applications:
 - น้ำฟู กระดูก** (สถานะ: ดำเนินการ) ผู้ใช้งาน: toto อีเมล: toto@gmail.com สถานที่: 7/10/2568
 - ปิดชั่วคราว**
 - ถอนสิทธิ์ Seller**
 - seller2** (สถานะ: ดำเนินการ) ผู้ใช้งาน: seller2 อีเมล: seller2@gmail.com สถานที่: 8/10/2568
 - ปิดชั่วคราว**
 - ถอนสิทธิ์ Seller**
 - Nicky_SellMan** (สถานะ: ดำเนินการ) ผู้ใช้งาน: NickyAnp อีเมล: nickyanp@gmail.com สถานที่: 8/10/2568
 - ปิดชั่วคราว**
 - ถอนสิทธิ์ Seller**

ภาพที่ 47 หน้าการจัดการร้านค้า (ต่อ) ในส่วนที่สอง จะเป็นการจัดการร้านค้า โดยที่ผู้ดูแลระบบจะสามารถ ‘ปิดชั่วคราว’ หรือ ‘ถอนสิทธิ์ seller’

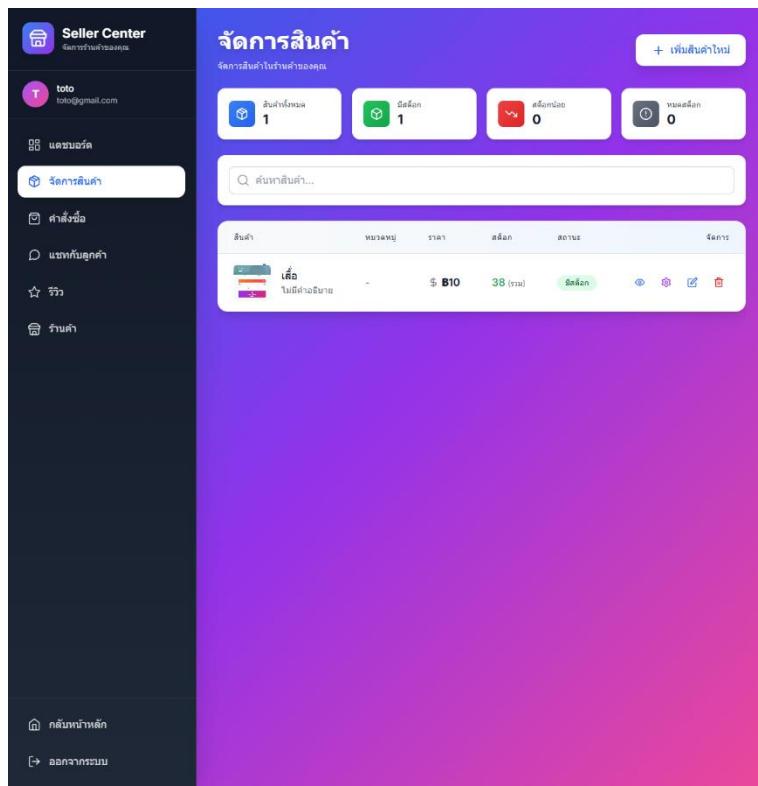


ภาพที่ 48 หน้าการตั้งค่า ในหน้านี้ผู้ดูแลระบบจะสามารถตั้งค่าเกี่ยวกับแพลตฟอร์มได้

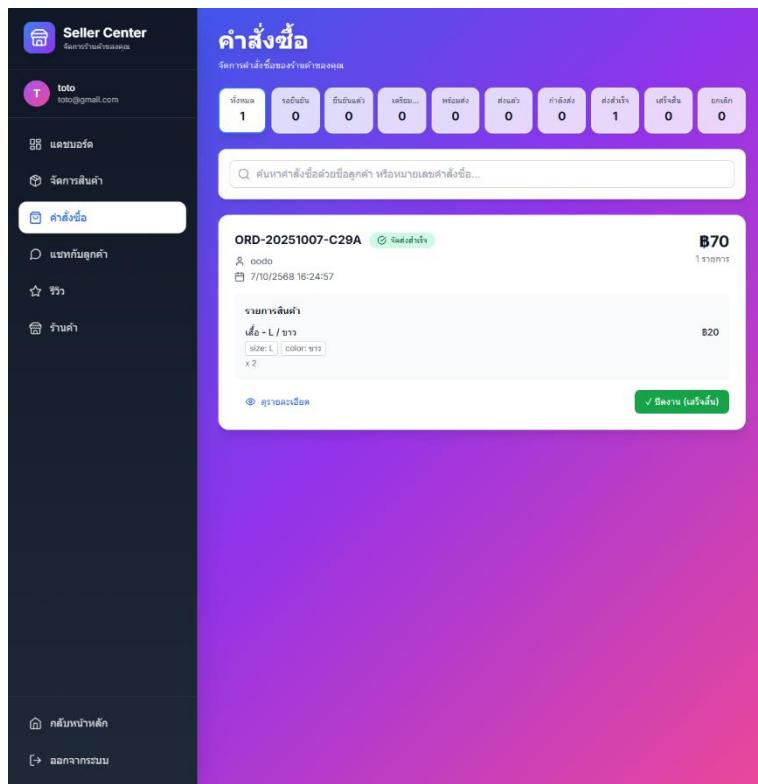
4.3 ส่วนของผู้ขาย (Seller)



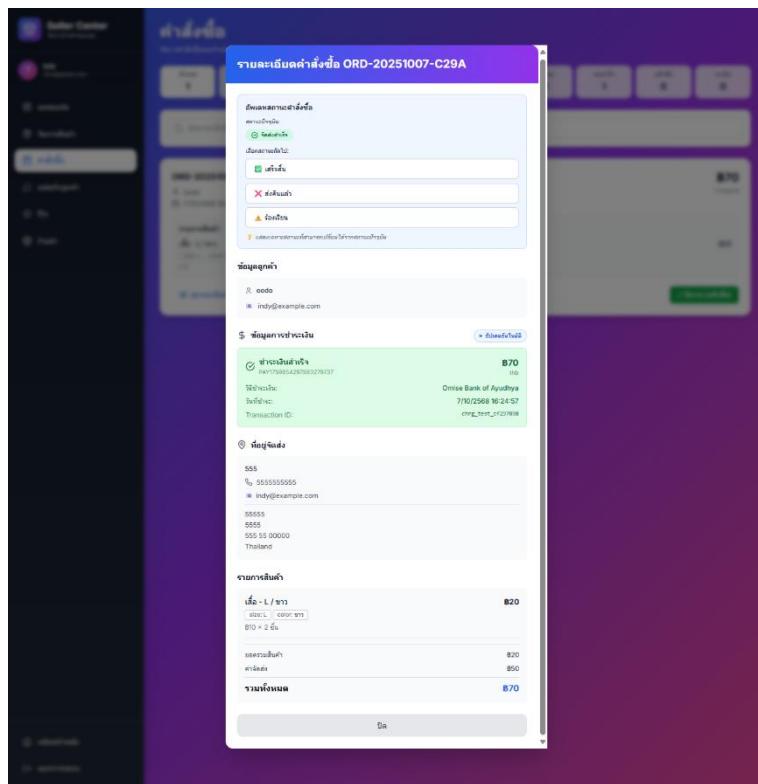
ภาพที่ 49 หน้าแดชบอร์ดของผู้ขาย โดยเป็นหน้าภาพรวมของร้านค้า แสดงสถิติและข้อมูลสำคัญ



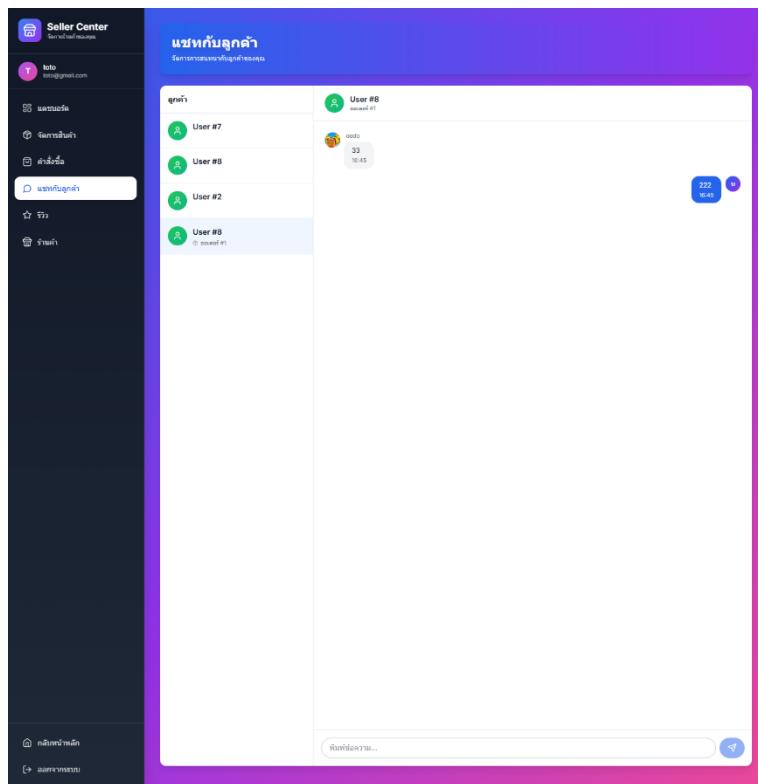
ภาพที่ 50 หน้าจัดการสินค้า โดยเป็นหน้าสำหรับจัดการสินค้าทั้งหมดในร้าน จะสามารถดูรายการสินค้าทั้งหมด เพิ่มสินค้า แก้ไขข้อมูลสินค้า และลบสินค้า นอกจากนี้ยังสามารถค้นหาสินค้าได้เช่นกัน



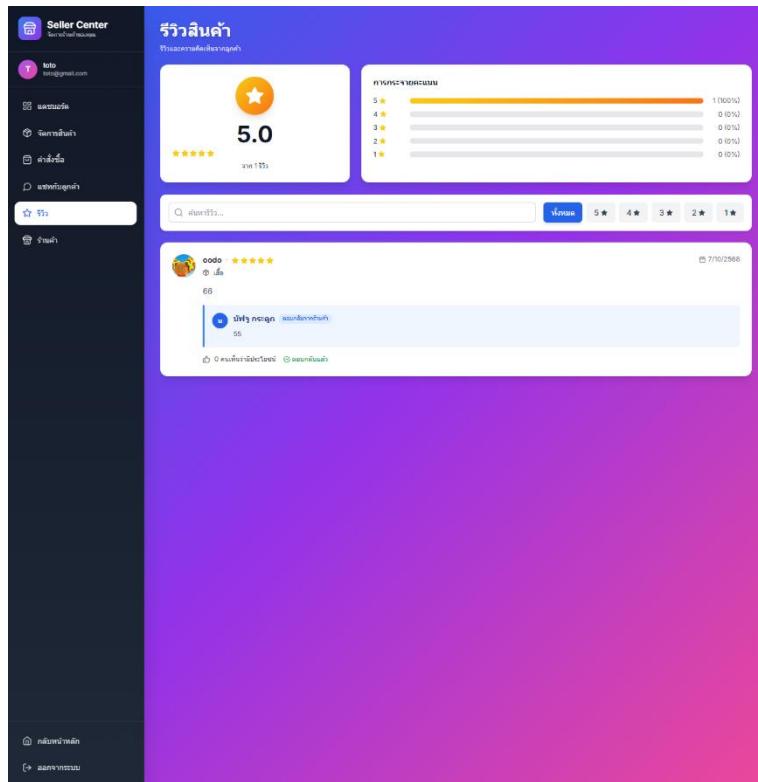
ภาพที่ 51 หน้าจัดการคำสั่งซื้อ ในหน้านี้จะเป็นหน้าแสดงและจัดการคำสั่งซื้อทั้งหมดของร้าน



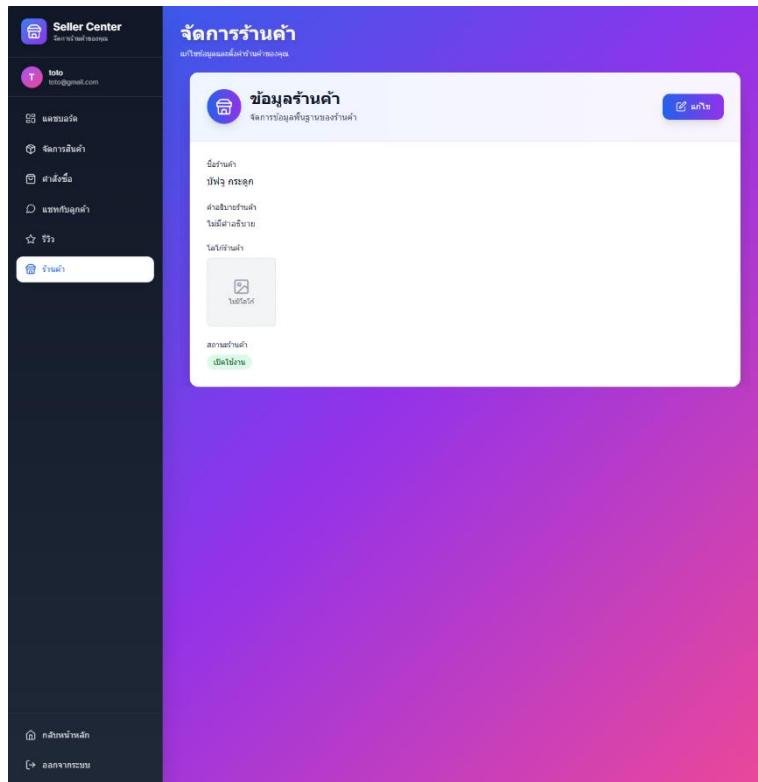
ภาพที่ 52 หน้ารายละเอียดคำสั่งซื้อ เป็นหน้าแสดงข้อมูลคำสั่งซื้อแบบละเอียด สามารถดูได้จากการคลิกดูรายละเอียดในหน้าจัดการคำสั่งซื้อ



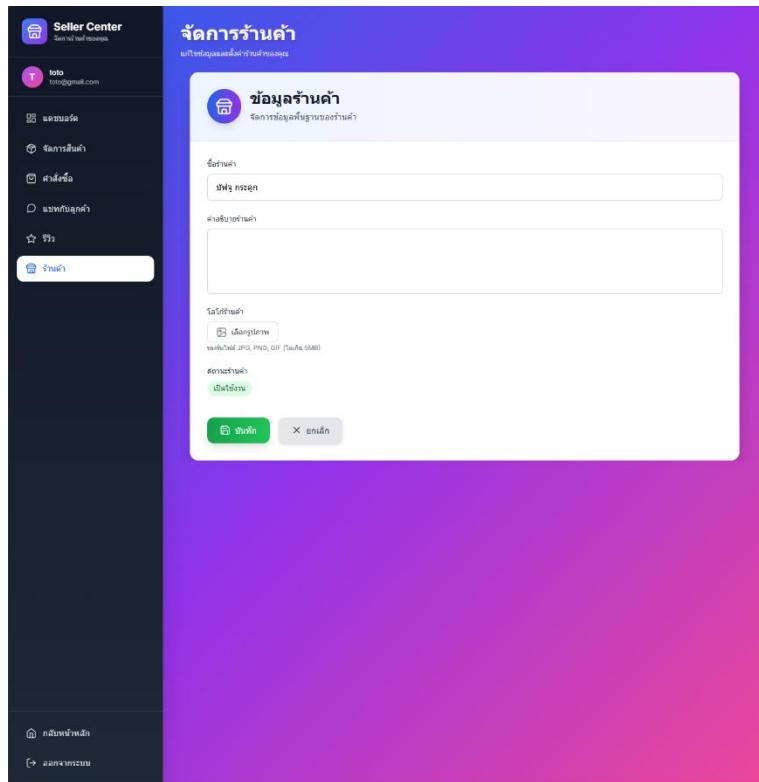
ภาพที่ 53 หน้าแดชบอร์ดของผู้ขาย เป็นส่วนของการติดต่อกับผู้ซื้อ



ภาพที่ 54 หน้าการจัดการรีวิว ในหน้านี้ผู้ขายสามารถดูคะแนนรีวิวของค้า การกระจายของคะแนน ในหน้านี้จะสามารถค้นหาข้อความรีวิว และกรองรีวิวจากคะแนนได้



ภาพที่ 55 หน้าการจัดการร้านค้า ในหน้านี้ผู้ขายสามารถแก้ไขข้อมูลของร้านได้



ภาพที่ 56 หน้าการจัดการร้านค้า (ต่อ) เมื่อกด ‘แก้ไข’ จะแสดงฟอร์มของการแก้ไขข้อมูล

บทที่ 5

อภิปรายและข้อเสนอแนะ

จากการพัฒนาแพลตฟอร์มสำหรับการซื้อขายผ่านช่องทางออนไลน์ พบว่าระบบสามารถนำมารูปอภิปรายผลและข้อเสนอแนะได้ตามหัวข้อดังต่อไปนี้

- 5.1 สรุปผลการดำเนินโครงการ
- 5.2 ปัญหาและข้อจำกัด
- 5.3 ข้อเสนอแนะ

5.1 สรุปผลการดำเนินโครงการ

โครงการพัฒนาเว็บไซต์ E-commerce นี้ได้ดำเนินการสำเร็จตามวัตถุประสงค์ที่กำหนดไว้ โดยมีผลการดำเนินงานที่สำคัญดังนี้

5.1.1 ด้านการพัฒนาระบบ

ระบบได้รับการพัฒนาครบถ้วนตามขอบเขตที่กำหนด โดยแบ่งเป็น 3 บทบาทหลัก ได้แก่ ผู้ซื้อ ผู้ขาย และผู้ดูแลระบบ ระบบสำหรับผู้ซื้อประกอบด้วยฟังก์ชันการเลือกซื้อสินค้า การรีวิวสินค้า การชำระเงินออนไลน์ การแท็กบัญชี การจัดการโปรไฟล์ และการเลือกรายการสินค้าโปรด ระบบสำหรับผู้ขายมีความสามารถในการจัดการสินค้า จัดการคำสั่งซื้อ การแท็กบัญชี จัดการการรีวิว และระบบการเงิน ส่วนระบบสำหรับแอดมิน สามารถจัดการการแบบร้านค้าและจัดการบัญชีผู้ใช้งาน 3 บทบาทได้

5.1.2 ด้านเทคโนโลยีและการเรียนรู้

การพัฒนาโครงงานนี้ทำให้ได้เรียนรู้และฝึกฝนการใช้งานเทคโนโลยีที่ทันสมัยอย่างครบถ้วน ในส่วน Frontend ได้ใช้ Next.js ร่วมกับ TypeScript ทำให้โค้ดมีความปลอดภัยและง่ายต่อการดีบัก การใช้ Tailwind CSS ช่วยให้การออกแบบ UI เป็นไปอย่างรวดเร็วและตอบสนอง ในส่วน Backend การใช้ Spring Boot ช่วยให้การพัฒนา API มีประสิทธิภาพ PostgreSQL ให้ความมั่นคงในการจัดเก็บข้อมูล JPA ช่วยในการจัดการฐานข้อมูลอย่างเป็นระบบ และ JWT ทำให้ระบบมีความปลอดภัยในการยืนยันตัวตน

5.1.3 ด้านหลักการและแนวทาง

โครงงานนี้เน้นการศึกษาและประยุกต์ใช้หลักการ CRUD Operations อย่างถูกต้องครบถ้วนในทุกฟังก์ชันของระบบ การนำ SOLID Principles มาใช้ในการออกแบบและเขียนโค้ดทำให้ระบบมีโครงสร้างที่ชัดเจน ง่ายต่อการดูแลรักษา และสามารถขยายฟีเจอร์ใหม่ได้ในอนาคต การจัดการฐานข้อมูลด้วย ER Diagram และ Relational Model ช่วยให้การออกแบบฐานข้อมูลมีความสัมพันธ์ที่ถูกต้องและมีประสิทธิภาพ

5.1.4 ด้านการทำงานเป็นทีม

การใช้ Git และ GitHub เป็นระบบควบคุมเวอร์ชันช่วยให้การทำงานเป็นทีมเป็นไปอย่างราบรื่น สามารถติดตามการเปลี่ยนแปลงของโค้ด และแก้ไขปัญหา Conflict ได้อย่างมีประสิทธิภาพ การใช้ Postman ในการทดสอบ API ช่วยให้การพัฒนาและ Debug เป็นไปอย่างเป็นระบบ และ pgAdmin ช่วยในการจัดการฐานข้อมูลแบบ GUI ทำให้ง่ายต่อการตรวจสอบและแก้ไขข้อมูล

5.1.5 ผลลัพธ์ที่ได้รับ

ระบบที่พัฒนาขึ้นสามารถใช้งานได้จริงและมีความเสถียร รองรับการใช้งานของผู้ใช้หลายบทบาทพร้อมกัน มีระบบความปลอดภัยที่เหมาะสม และมีโครงสร้างที่สามารถขยายและปรับปรุงได้ในอนาคต โครงการนี้จึงบรรลุวัตถุประสงค์ทั้ง 3 ข้อ คือ การศึกษาหลักการพัฒนาเว็บไซต์ E-commerce การฝึกฝนการใช้งานเทคโนโลยี และการพัฒนาแพลตฟอร์มที่สามารถขยายได้

5.2 ปัญหาและข้อจำกัด

5.2.1 ปัญหาด้านเทคนิค

1) การทดสอบระบบแบบ Real-time

ระบบแข็งแกร่งกว่าผู้ชี้อและผู้ขายต้องการการทำงานแบบ Real-time ซึ่งการทดสอบและการจัดการ WebSocket หรือ Long Polling มีความซับซ้อนและต้องใช้เวลาในการแก้ปัญหา

5.2.2 ข้อจำกัดด้านฟังก์ชันการทำงาน

1) ระบบการชำระเงิน

ระบบการชำระเงินออนไลน์ในปัจจุบันอาจยังไม่ครบถ้วนเท่าที่ควร การเขื่อมต่อ กับ Payment Gateway จะต้องมีการสมัครและตรวจสอบที่ซับซ้อน ซึ่งอาจใช้เป็น Mock-up ในระหว่างการพัฒนา

2) ระบบแนะนำสินค้า

ระบบยังไม่มีฟีเจอร์การแนะนำสินค้าตาม Machine Learning หรือ AI ซึ่งเป็นฟีเจอร์ที่สำคัญในแพลตฟอร์ม E-commerce อย่างไร การแนะนำสินค้าในปัจจุบันอาจใช้เพียงการเรียนตามความนิยมหรือการค้นหาพื้นฐาน

3) ระบบการจัดส่งและติดตามพัสดุ

ระบบยังไม่มีการเขื่อมต่อ กับบริษัทขนส่งจริง การติดตามสถานะพัสดุอาจทำได้เพียงในระดับพื้นฐาน ไม่สามารถติดตาม Real-time Location ของพัสดุได้

4) ระบบการจัดการคลังสินค้า

ระบบยังไม่มีการจัดการคลังสินค้าที่ซับซ้อน เช่น การแจ้งเตือนเมื่อสินค้าใกล้หมด

5.2.3 ข้อจำกัดด้านเวลาและทรัพยากร

1) ระยะเวลาในการพัฒนา

เนื่องจากโครงการมีกรอบเวลาเพียง 3 เดือน (สิงหาคม-ตุลาคม) ทำให้บางฟีเจอร์ที่ต้องการเวลาในการพัฒนานาน เช่น ระบบ Analytics ที่ซับซ้อน หรือระบบ Notification อาจไม่สามารถพัฒนาได้ทัน

2) การทดสอบ Load Testing

เนื่องจากข้อจำกัดด้านทรัพยากร การทดสอบระบบภายใต้ภาระสูง (Load Testing) หรือการทดสอบกับผู้ใช้งานจำนวนมากพร้อมกันอาจทำได้จำกัด ทำให้ไม่สามารถประเมินประสิทธิภาพที่แท้จริงของระบบได้

5.2.4 ข้อจำกัดด้านความปลอดภัย

1) การตรวจสอบความปลอดภัยอย่างครบถ้วน

แม้ระบบจะมี JWT และมาตรการรักษาความปลอดภัยพื้นฐาน แต่อาจยังไม่ได้ผ่านการตรวจสอบความปลอดภัยอย่างละเอียดด้วย Penetration Testing หรือ Security Audit จากผู้เชี่ยวชาญ

2) การจัดการข้อมูลส่วนบุคคล (PDPA)

ระบบอาจยังไม่มีการจัดการข้อมูลส่วนบุคคลที่เป็นไปตาม พ.ร.บ. คุ้มครองข้อมูลส่วนบุคคล อย่างครบถ้วน เช่น การขอความยินยอม การลบข้อมูล หรือการส่งออกข้อมูลส่วนบุคคล

5.3 ข้อเสนอแนะ

5.3.1 ข้อเสนอแนะสำหรับการพัฒนาต่อไป

1) การเพิ่มระบบแนะนำสินค้าด้วย AI และ Machine Learning

ควรพัฒนาระบบแนะนำสินค้าที่ใช้ Machine Learning เพื่อวิเคราะห์พฤติกรรมของผู้ใช้ ประวัติการซื้อ และสินค้าที่เคยดู เพื่อแนะนำสินค้าที่เหมาะสมกับผู้ใช้แต่ละคน ซึ่งจะช่วยเพิ่มโอกาสในการขายและสร้างประสบการณ์ที่ดีให้กับผู้ใช้

2) การพัฒนา Mobile Application

ควรพัฒนาแอปพลิเคชันบนมือถือทั้งระบบ iOS และ Android เพื่อเพิ่มความสะดวกในการใช้งานและเข้าถึงผู้ใช้ได้มากขึ้น โดยอาจใช้ React Native หรือ Flutter ในการพัฒนาเพื่อลดเวลาและต้นทุน

3) การเชื่อมต่อกับระบบ Payment Gateway จริง

ควรเชื่อมต่อกับ Payment Gateway ที่เป็นที่นิยมในประเทศไทย เช่น PromptPay, TrueMoney Wallet, หรือบัตรเครดิต/เดบิต ผ่าน Omise หรือ 2C2P เพื่อให้ระบบสามารถใช้งานได้จริงในเชิงพาณิชย์

4) การเพิ่มระบบ Multi-language และ Multi-currency

หากต้องการขยายตลาดไปยังต่างประเทศ ควรเพิ่มการรองรับหลายภาษาและหลายสกุลเงิน รวมถึงการคำนวณอัตราแลกเปลี่ยนแบบ Real-time

5) การพัฒนาระบบ Loyalty Program และ Gamification

ควรเพิ่มระบบสะสมคะแนน คูปอง โปรโมชั่น และการเล่นเกมภายในแพลตฟอร์มเพื่อเพิ่มการมีส่วนร่วมและความภักดีของลูกค้า

5.3.2 ข้อเสนอแนะด้านเทคนิค

1) การใช้ Microservices Architecture

หากระบบมีการขยายใหญ่ขึ้น ควรพิจารณาแยกระบบออกเป็น Microservices เช่น แยก Service สำหรับการจัดการผู้ใช้ Service สำหรับสินค้า Service สำหรับคำสั่งซื้อ เป็นต้น เพื่อให้ง่ายต่อการขยายและบำรุงรักษา

2) การใช้ CDN สำหรับภาพและไฟล์ Static

ควรใช้ Content Delivery Network (CDN) เช่น Cloudflare หรือ AWS CloudFront เพื่อเพิ่มความเร็วในการโหลดภาพสินค้าและไฟล์ Static ทำให้เว็บไซต์มีประสิทธิภาพดีขึ้น

3) การทำ Caching อย่างมีประสิทธิภาพ

ควรใช้ Redis หรือ Memcached สำหรับ Caching ข้อมูลที่ถูกเรียกใช้บ่อย เช่น ข้อมูลสินค้า ข้อมูลผู้ใช้ เพื่อลดภาระของฐานข้อมูลและเพิ่มความเร็วในการตอบสนอง

4) การทำ API Rate Limiting

ควรเพิ่มระบบจำกัดจำนวนการเรียก API (Rate Limiting) เพื่อป้องกันการโจมตีแบบ DDoS และการใช้งานที่เกินความจำเป็น

5.3.3 ข้อเสนอแนะด้านประสบการณ์ผู้ใช้

1) การปรับปรุง UI/UX อย่างต่อเนื่อง

ควรทำการทดสอบกับผู้ใช้จริง (User Testing) และรวบรวม Feedback เพื่อปรับปรุง UI/UX ให้ใช้งานง่ายและสร้างประสบการณ์ที่ดีมากขึ้น

2) การเพิ่มระบบแจ้งเตือนแบบ Push Notification

ควรเพิ่มระบบแจ้งเตือนแบบ Push เพื่อแจ้งเตือนผู้ใช้เกี่ยวกับสถานะคำสั่งซื้อ โปรโมชั่น หรือข้อความใหม่จากผู้ขาย

3) การเพิ่มระบบ Voice Search

ควรพัฒนาเพิ่มฟีเจอร์การค้นหาด้วยเสียง เพื่อให้ผู้ใช้สามารถค้นหาสินค้าได้สะดวกมากขึ้น โดยเฉพาะบนอุปกรณ์มือถือ

4) การพัฒนาระบบ AR (Augmented Reality)

สำหรับสินค้าบางประเภท เช่น เฟอร์นิเจอร์ เครื่องสำอาง ควรพัฒนาระบบ AR ที่ให้ผู้ใช้สามารถดูสินค้าในสภาพแวดล้อมจริงก่อนตัดสินใจซื้อ

5.3.4 ข้อเสนอแนะด้านความปลอดภัย

1) การทำ Penetration Testing

ควรจ้างผู้เชี่ยวชาญด้านความปลอดภัยมาทำการทดสอบเจาะระบบเพื่อค้นหาช่องโหว่ที่อาจเกิดขึ้นและแก้ไขก่อนที่จะเปิดให้บริการจริง

2) การทำ Security Audit อย่างสม่ำเสมอ

ควรมีการตรวจสอบความปลอดภัยของระบบอย่างสม่ำเสมอ และอัพเดท Dependencies ต่างๆ เพื่อป้องกันช่องโหว่ที่อาจเกิดจากเวอร์ชันเก่า

បរណ្ណកម្ម

- AWS. (2025). Cross-Origin Resource Sharing (CORS). Amazon Web Services.
<https://aws.amazon.com/what-is/cross-origin-resource-sharing>
- Bauer, C., & King, G. (2007). Java Persistence with Hibernate. Manning Publications.
<https://www.manning.com/books/java-persistence-with-hibernate>
- Bierman, G., Abadi, M., & Torgersen, M. (2014). Understanding TypeScript. In ECOOP 2014 – Object-Oriented Programming (pp. 257–281). Springer.
https://link.springer.com/chapter/10.1007/978-3-662-44202-9_11
- Cherny, B. (2019). Programming TypeScript: Making your JavaScript applications scale. O'Reilly Media. <https://www.oreilly.com/library/view/programming-typescript/9781492037644>
- Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures [Doctoral dissertation, University of California, Irvine].
<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- Fielding, R. T., & Taylor, R. N. (2002). Principled design of the modern web architecture. ACM Transactions on Internet Technology, 2(2), 115–150.
<https://doi.org/10.1145/514183.514185>
- Fowler, M. (2002). Patterns of enterprise application architecture. Addison-Wesley.
<https://martinfowler.com/books/eaa.html>
- Fowler, M. (2004). Inversion of control containers and the dependency injection pattern. Martin Fowler. <https://martinfowler.com/articles/injection.html>
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design patterns: Elements of reusable object-oriented software. Addison-Wesley. <https://www.pearson.com/store/p/design-patterns-elements-of-reusable-object-oriented-software/P100000296097>
- Gupta, A. (2016). Learning Spring Boot. Packt Publishing.
<https://www.packtpub.com/product/learning-spring-boot/9781784393021>
- Johnson, R., Hoeller, J., Arendsen, A., Risberg, T., & Davison, D. (2009). Professional Java development with the Spring Framework. Wrox Press. <https://www.wiley.com/en-us/Professional+Java+Development+with+the+Spring+Framework-p-9780764574832>
- Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT) (RFC 7519). Internet Engineering Task Force (IETF). <https://tools.ietf.org/html/rfc7519>
- JWT.io. (2024). Introduction to JSON Web Tokens. <https://jwt.io/introduction>
- Keith, M., & Stafford, M. (2009). Pro JPA 2: Mastering the Java Persistence API. Apress.
<https://www.apress.com/gp/book/9781430219569>

- Leff, A., & Rayfield, J. T. (2001). Web-application development using the Model/View/Controller design pattern. In Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference (pp. 118-127). IEEE. <https://doi.org/10.1109/EDOC.2001.950428>
- Liskov, B. (1987). Keynote address - Data abstraction and hierarchy. ACM SIGPLAN Notices, 23(5), 17-34. <https://doi.org/10.1145/62139.62141>
- Martin, R. C. (2000). Design principles and design patterns. Object Mentor, 1(34), 597. https://web.archive.org/web/20150906155800/http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf
- Martin, R. C. (2002). Agile software development: Principles, patterns, and practices. Prentice Hall. <https://www.pearson.com/store/p/agile-software-development-principles-patterns-and-practices/P100000147019>
- Martin, R. C. (2003). The single responsibility principle. Object Mentor. <https://web.archive.org/web/20150202200348/http://www.objectmentor.com/resources/articles/srp.pdf>
- Martin, R. C. (2008). Clean code: A handbook of agile software craftsmanship. Prentice Hall. <https://www.pearson.com/store/p/clean-code-a-handbook-of-agile-software-craftsmanship/P100000148329>
- Massé, M. (2011). REST API design rule book. O'Reilly Media. <https://www.oreilly.com/library/view/rest-api-design/9781449317904>
- MDN Web Docs. (2025). Cross-Origin Resource Sharing (CORS). Mozilla Developer Network. <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
- Microsoft. (2012). TypeScript: JavaScript that scales. Microsoft Corporation. <https://www.typescriptlang.org>
- Mishra, A., & Kalra, S. (2019). Analysis of security issues in JSON Web Token. In 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 27-32). IEEE. <https://doi.org/10.1109/ICCCIS48478.2019.8974493>
- Next.js Documentation. (2023). Getting started with Next.js. Vercel. <https://nextjs.org/docs>
- Obe, R., & Hsu, L. (2014). PostgreSQL: Up and running (2nd ed.). O'Reilly Media. <https://www.oreilly.com/library/view/postgresql-up-and/9781449373184>
- OWASP. (2025). Cross-Origin Resource Sharing (CORS). Open Web Application Security Project. https://owasp.org/www-community/attacks/CORS_OriginHeaderScrutiny
- PortSwigger. (2025). Cross-origin resource sharing (CORS). PortSwigger Web Security. <https://portswigger.net/web-security/cors>
- Reenskaug, T. (1979). The original MVC reports. Xerox PARC. <https://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>

- Richardson, L., & Ruby, S. (2007). RESTful web services. O'Reilly Media.
<https://www.oreilly.com/library/view/restful-web-services/9780596529260>
- Stonebraker, M., & Rowe, L. A. (1986). The design of POSTGRES. ACM SIGMOD Record, 15(2), 340-355. <https://doi.org/10.1145/16856.16888>
- Suryawanshi, S. (2023). CORS vulnerability: What it is and how to prevent it. Indusface Blog.
<https://www.indusface.com/blog/cors-vulnerability>
- Tailwind Labs. (2019). Tailwind CSS documentation. Tailwind Labs Inc.
<https://tailwindcss.com/docs>
- Tenable. (2023). Cross-Origin Resource Sharing (CORS) security issues. Tenable, Inc.
<https://www.tenable.com/plugins/nessus/109703>
- Vaadata. (2025). Understanding JSON Web Tokens (JWT). Vaadata Security.
<https://vaadata.com/blog/understanding-json-web-tokens-jwt>
- Vercel. (2021). Next.js: The React Framework for production. Vercel Inc. <https://nextjs.org>
- Walls, C. (2016). Spring Boot in action. Manning Publications.
<https://www.manning.com/books/spring-boot-in-action>
- Wikipedia. (2025). Cross-origin resource sharing. Wikimedia Foundation.
https://en.wikipedia.org/wiki/Cross-origin_resource_sharing