ชื่อโครงงาน ตามใจเชฟ

โดย

นางสาวกมลนิตย์ ธรรมชาติ รหัสนักศึกษา 65070008

นางสาวกานต์พิชชา เนียมศรี รหัสนักศึกษา 65070017

นางสาวชาลิสา สันติธัญญาโชค รหัสนักศึกษา 65070050

นางสาวฐิตารีย์ รวิชุติวรรณ รหัสนักศึกษา 65070057

นางสาวปุรินปรัชญ์ อาษานอก รหัสนักศึกษา 65070088

เสนอ

ดร. พัฒนพงษ์ ฉันทมิตรโอภาส

ผศ.ดร. ลภัส ประดิษฐ์ทัศนีย์

โครงานนี้เป็นส่วนหนึ่งของวิชาเทคโนโลยีกลุ่มเมฆ

แขนงวิชา การพัฒนาสื่อประสมและเกม

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2567

คำนำ

การประมวลผลแบบกลุ่มเมฆ (Cloud Computing) เป็นหนึ่งในเทคโนโลยีที่ได้รับความนิยมอย่างมากในปัจจุบัน เนื่องจากความสามารถในการจัดการข้อมูลและการประมวลผลอย่างมีประสิทธิภาพ โดยไม่จำเป็นต้องพึ่งพาทรัพยากรฮาร์ดแวร์ที่มีอยู่ในสถานที่เดียวกัน โครงงานนี้จัดทำขึ้นเพื่อประกอบการเรียนการสอนวิชาการประมวลผลแบบกลุ่มเมฆ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ในโครงงานนี้ผู้จัดทำได้พัฒนาเว็บแอปพลิเคชันสำหรับร้านอาหารโอมากาเสะ โดยใช้บริการคลาวด์จาก Amazon Web Services (AWS) ซึ่งเป็นตัวอย่างที่ดีของการนำเทคโนโลยีคลาวด์มาใช้ในการสร้างระบบที่ตอบสนองต่อความต้องการในด้านการจัดการเมนูอาหาร, การจัดการคอร์ส, ระบบจอง, ระบบชำระเงิน, และระบบตรวจสอบการจอง

หากโครงการนี้มีข้อผิดพลาดประการใด ทางคณะผู้จัดทำขออภัยมา ณ ที่นี้ และหวังเป็นอย่างยิ่งว่าข้อมูลและแนวทางที่นำเสนอในโครงงานนี้จะเป็นประโยชน์และสามารถนำไปพัฒนาต่อได้ในอนาคต

คณะผู้จัดทำ

COPYRIGHT 2024

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ใบรับรองโครงงาน (PROJECT)

เรื่อง ตามใจเชฟ

TamjaiChef

นางสาวกมลนิตย์ ธรรมชาติ รหัสนักศึกษา 65070008

นางสาวกานต์พิชชา เนียมศรี รหัสนักศึกษา 65070017

นางสาวชาลิสา สันติธัญญาโชค รหัสนักศึกษา 65070050

นางสาวฐิตารีย์ รวิชุติวรรณ รหัสนักศึกษา 65070057

นางสาวปุรินปรัชญ์ อาษานอก รหัสนักศึกษา 65070088

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด

รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ

การศึกษาวิชาเทคโนโลยีกลุ่มเมฆ แขนงวิชาการพัฒนาสื่อประสมและเกม

คณะเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2567

คณะผู้จัดทำ

หัวข้อโครงงาน ตามใจเชฟ

นักศึกษา นางสาวกมลนิตย์ ธรรมชาติ, นางสาวกานต์พิชชา เนียมศรี, นางสาวชาลิสา สันติธัญญาโชค , นางสาวฐิตารีย์ รวิชุติวรรณ ,นางสาวปุรินปรัชญ์ อาษานอก

แขนงวิชา การพัฒนาสื่อประสมและเกม

ปีการศึกษา 1/2567

บทคัดย่อ

อาหารที่เป็นปัจจัยสี่ในการดำรงชีวิตที่ไม่สามารถขาดได้ ด้วยความสำคัญที่ส่งผลต่อการดำรงชีวิตนี้ทำให้อาหารกลายเป็นศิลปะที่มีการปรุงแต่งทั้งรสชาติ สี และมีรูปแบบที่หลากหลายอันเกิดจากวัฒนธรรมของแต่ละประเทศ แต่ละภูมิภาคจนกลายมาเป็น “โอมากาเสะ” ซึ่งเป็นอาหารที่ใช้ความละเอียดพิถีพิถัน การใส่ใจในแต่ละขั้นตอน โดยเชฟจะเป็นผู้รังสรรค์อาหารในแต่ละจานขึ้นมาตามวัตถุดิบที่เปลี่ยนไปตามฤดูกาล ซึ่งพ้องกับคำว่า “โอมากาเสะ” ที่แปลว่าการเสิร์ฟแบบ “ตามใจเชฟ” โดยแอปพลิเคชันจะช่วยอำนวยความสะดวกและลดเวลาในขั้นตอนการจอง การตรวจสอบการชำระเงิน และการจัดการกับเมนูอาหารในแต่ละวันได้

แอปพลิเคชันตามใจเชฟนั้นเล็งเห็นถึงความสำคัญของการรับประทานอาหารและความสะดวกรวดเร็วในการบริการทั้งฝั่งของร้านอาหารและลูกค้า โดยในฝั่งลูกค้านั้นจะสามารถทำการชำระเงินและตรวจสอบคอร์สที่จองเอาไว้ได้ ในส่วนของทางร้านนั้นจะสามารถจัดการกับรายการอาหารทั้งหมดภายในร้าน จัดการเมนูอาหาร ดูรายการทั้งหมดที่ลูกค้าจอง ชำระเงินมัดจำ ชำระเงินสุทธิ และยกเลิกการจอง ซึ่งได้มีการนำเอาเทคโนโลยีการประมวลผลแบบกลุ่มเมฆ (Cloud) มาใช้สำหรับเซิร์ฟเวอร์ ดูแลจัดการฐานข้อมูลทั้งการจัดเก็บไฟล์และการสำรองข้อมูลที่จำเป็น การจัดการด้านความปลอดภัยโดยการกำหนดการเข้าถึงของแอดมินเพื่อให้มั่นใจว่าข้อมูลของร้านและลูกค้าจะถูกดูแลอย่างดี

กิตติกรรมประกาศ

คณะผู้จัดทำได้เลือกหัวข้อนี้ในการทำงานโครงงาน เนื่องจากความสนใจในด้านอาหารของคณะผู้จัดทำ โครงงาน “ตามใจเชฟ” นี้สำเร็จลุล่วงไปได้ด้วยดีนี้เป็นผลมาจากการให้คำปรึกษาที่ดี และคอยติดตามผลอยู่ตลอดจากอาจารย์พัฒนพงษ์ ฉันทมิตรโอภาส และอาจารย์ลภัส ประดิษฐ์ทัศนีย์ ทั้งยังให้ความรู้แนวทางในการพัฒนา และแนวทางการแก้ไขปัญหาที่ถูกจุดทำให้โครงงานสามารถดำเนินงานไปได้อย่างรวดเร็ว

ทั้งนี้จะขาดเพื่อนๆ ของคณะผู้จัดทำไปไม่ได้เลยที่คอยให้กำลังใจและช่วยเหลือให้การทำโครงงานครั้งนี้ผ่านพ้นไปได้โดยดี

สารบัญ

บทที่ 1 1

บทนำ 1

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของระบบ 2

1. วัตถุประสงค์ของระบบ 2

2. ฟังก์ชันหลัก 2

3. ข้อกำหนดด้านเทคนิค 2

4. ข้อจำกัดของระบบ 2

5. ความต้องการด้านความปลอดภัย 3

6. ข้อกำหนดด้านการใช้งาน 3

1.4 ประโยชน์ที่คาดว่าจะได้รับ 3

1.5 โครงสร้างของรายงาน 4

บทที่ 2 5

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง 5

1. ทฤษฎีที่เกี่ยวข้อง 5

1.3 Web Application 7

1.4 Amazon Web Services (AWS) 8

2. เทคโนโลยีที่เกี่ยวข้อง 8

2.1 สถาปัตยกรรม อเมซอน คลาวด์เทคโนโลยี 8

2.2 สถาปัตยกรรมเครือข่าย 11

2.3 สถาปัตยกรรมฐานข้อมูล 11

2.4 เทคโนโลยีที่ใช้ 12
บทที่ 3 16
3. ขั้นตอนการดำเนินงานของโครงงาน 16
3.1 ความต้องการของระบบ 16
3.2 การวิเคราะห์และออกแบบระบบ 16
3.3 หลักการทำงานของระบบ 17
บทที่ 4 36
ระบบต้นแบบ 36
4.1 ส่วนติดต่อผู้ใช้งาน 36
4.2 ผลการดำเนินงานของการใช้งาน AWS ร่วมกับเว็บแอปพลิเคชัน 43
บทที่ 5 49
สรุปผลการดำเนินงาน 49
5.1 ที่มาและการทำงานโดยรวมของแอปพลิเคชัน 49
5.2 ประโยชน์ของแอปพลิเคชัน 49
5.3 ปัญหาและอุปสรรคที่เกิดขึ้นในการพัฒนา 49
5.4 ข้อจำกัดระบบ 50
5.5 ประมาณการค่าใช้จ่าย 50
5.6 การวิเคราะห์และออกแบบเพิ่มเติมโดยไม่คำนึงถึงค่าใช้จ่าย 51

สารบัญรูป

รูปที่ 2.1 ภาพประกอบ Role-Based Access Control 6
รูปที่ 2.2 ภาพประกอบ Web Application 7
รูปที่ 2.3 Amazon Web Services Logo 8
รูปที่ 2.4 Amazon EC2 Logo 9
รูปที่ 2.5 ALB Load Balancer 9
รูปที่ 2.6 Auto Scaling 10
รูปที่ 2.7 Cloud 9 10
รูปที่ 2.8 Amazon VPC 11
รูปที่ 2.9 Amazon Relational Database Service 11
รูปที่ 2.13 HyperText Markup Language (HTML) Logo 12
รูปที่ 2.14 Cascading Style Sheets (CSS) Logo 12
รูปที่ 2.15 JavaScript Logo 13
รูปที่ 2.17 XAMPP Logo 14
รูปที่ 2.18 SQLite Logo 15
รูปที่ 2.19 Figma Logo 15
รูปที่ 3.1 โครงสร้างการใช้งาน AWS 16
รูปที่ 3.2 Use Case Diagram 17
รูปที่ 3.3 เข้าสู่ระบบ 29
รูปที่ 3.4 ลงทะเบียน 29
รูปที่ 3.5 หน้าหลักระบบ 30
รูปที่ 3.6 ดูรายการเมนูอาหาร 31
รูปที่ 3.7 จองคอร์สโอมากาเสะ 31
รูปที่ 3.8 ชำระเงินมัดจำ 32

รูปที่ 3.9 ตรวจสอบการจองคอร์ส และ ยกเลิกการจอง 32
รูปที่ 3.10 ดูรายการการสั่งอาหาร 33
รูปที่ 3.11 ชำระเงิน 33
รูปที่ 3.12 แก้ไข เพิ่ม ลด เมนูอาหาร 34
รูปที่ 3.13 แก้ไข เพิ่ม ลดประเภทเมนู 34
รูปที่ 3.14 เพิ่มคอร์สโอมากาเสะ 35
รูปที่ 4.1 หน้าหลัก 36
รูปที่ 4.2 หน้าหลัก 37
รูปที่ 4.3 หน้าดูคอร์สอาหาร 37
รูปที่ 4.4 หน้า Log-In 38
รูปที่ 4.5 หน้าหลักลูกค้า 38
รูปที่ 4.6 หน้าจองคอร์สโอมากาเสะ(ลูกค้า) 39
รูปที่ 4.7 หน้าดูประวัติการจอง (ลูกค้า) 39
รูปที่ 4.8 หน้าหลักผู้จัดการ 40
รูปที่ 4.9 หน้าจัดการคอร์ส (ผู้จัดการ) 40
รูปที่ 4.10 หน้าจัดเมนู (ผู้จัดการ) 41
รูปที่ 4.11 หน้าเพิ่มคอร์ส (ผู้จัดการ) 41
รูปที่ 4.12 หน้าเพิ่มคอร์ส (ผู้จัดการ) 42
รูปที่ 4.13 Amazon Elastic Compute Cloud (Amazon EC2) 43
รูปที่ 4.14 Elastic Load Balancing 44
รูปที่ 4.15 Auto Scaling 44
รูปที่ 4.16 Auto Scaling 44
รูปที่ 4.17 Amazon Relational Database Service (Amazon RDS) 45
รูปที่ 4.18 Amazon Relational Database Service (Amazon RDS) 45
รูปที่ 4.19 Amazon Virtual Private Cloud (Amazon VPC) 46

รูปที่ 4.20 Amazon Virtual Private Cloud (Amazon VPC) 46

รูปที่ 4.21 Amazon Virtual Private Cloud (Route tables) 47

รูปที่ 4.21 Amazon Virtual Private Cloud (Security Groups) 47

รูปที่ 4.22 Amazon Cloud9 48

รูปที่ 4.23 Amazon Cloud9 48

รูปที่ 5.1 ประมาณการใช้จ่าย 50

รูปที่ 5.2 ประมาณการใช้จ่าย 50

สารบัญตาราง

ตารางที่ 3.1 Use Case Description จัดการรายการอาหารทั้งหมดในร้าน 19

ตารางที่ 3.2 Use Case Description จัดการคอร์สโอมากาเสะ 21

ตารางที่ 3.3 Use Case Description ดูรายการทั้งหมดที่ลูกค้าจอง 22

ตารางที่ 3.4 Use Case Description รับชำระเงินสุทธิ 24

ตารางที่ 3.5 Use Case Description จองคอร์สโอมากาเสะ 25

ตารางที่ 3.6 Use Case Description ชำระเงินมัดจำ 26

ตารางที่ 3.7 Use Case Description ยกเลิกการจอง 27

ตารางที่ 3.8 Use Case Description ตรวจสอบคอร์สโอมากาเสะที่จองไว้ 28

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

โอมากาเสะเป็นร้านอาหารที่มีความละเอียด ใส่ใจในขั้นตอนการทำอาหารและมีพิธีรีตอง สูง รายการอาหารในคอร์สโอมากาเสะจะเปลี่ยนแปลงไปตามแต่ละโอกาสหรือฤดูกาลที่เหมาะสม โดยทางร้านจะเป็นผู้ตัดสินใจเพียงฝ่ายเดียวว่าจะทำรายการอาหารอะไรบ้างระหว่างคอร์สนั้น เราเล็งเห็นว่าร้านโอมากาเสะมีรูปแบบร้านที่น่าสนใจในด้านการบริการที่มีแนวคิดในการ เสิร์ฟแบบ “ตามใจเชฟ” ซึ่งตรงกับความหมายของคำว่า “โอมากาเสะ” ในภาษาญี่ปุ่นนั่นเอง ขยายความคือ การรับประทานอาหารโดยที่ลูกค้าไม่ได้เลือกรายการเอง แต่ละรายการที่ เสิร์ฟเชฟจะเป็นคนจัดให้ โดยเราได้นำวิธีการนำเสนออาหารแบบโอมากาเสะและพัฒนาเพิ่มเติมเป็นรูปแบบของแอปพลิเคชัน ร้าน ที่สามารถควบคุมระบบต่างๆภายในร้านทั้งการจัดการรายการอาหารทั้งหมดในร้าน, จัดการ คอร์สโอมากาเสะ, ดูรายการทั้งหมดที่ลูกค้าจอง, รับชำระเงินสุทธิ, จองคอร์สโอมากาเสะ, ชำระเงินมัดจำ, ยกเลิกการจอง และให้ลูกค้าสามารถตรวจสอบคอร์สโอมากาเสะที่จองไว้ได้

1.2 วัตถุประสงค์

เพื่อศึกษาพื้นฐานและโครงสร้างของ Cloud Computing การออกแบบสถาปัตยกรรม Cloud ที่เหมาะสมเข้ากับโครงงาน เรียนรู้เทคโนโลยีและเครื่องมือที่ใช้ใน Cloud การจัดการความปลอดภัยและความมั่นคง อีกทั้งการทดสอบและการบำรุงรักษาระบบ โดยมีวัตถุประสงค์หลักดังนี้

1. เพื่อสร้างความเข้าใจในการปรับใช้ Cloud ในสถานการณ์จริง
2. เพื่อให้เกิดความรู้และความเข้าใจในวิชาที่ศึกษามากขึ้น
3. เพื่อเรียนรู้และศึกษาค้นคว้าความรู้เพิ่มเติมจากวิชาที่ศึกษา
4. เพื่อศึกษาแนวทางการรักษาความปลอดภัยในระบบ Cloud
5. เพื่อพัฒนาทักษะในการออกแบบสถาปัตยกรรมCloudและการออกแบบสถาปัตยกรรม Cloud ที่เหมาะสม

1.3 ขอบเขตของระบบ

1. วัตถุประสงค์ของระบบ

ระบบนี้มีวัตถุประสงค์เพื่อ นำเทคโนโลยีและเครื่องมือที่ใช้ในคลาวด์มาจัดการระบบร้านอาหาร โดยมุ่งเน้นที่ความสะดวกในการจัดการข้อมูล การเข้าถึงข้อมูล และการรักษาความปลอดภัยของข้อมูลร้านอาหารอย่างมีประสิทธิภาพ

2. ฟังก์ชันหลัก

- **ระบบจัดการรายการอาหารทั้งหมดในร้าน :** ระบบนี้เป็นการเพิ่ม ลบ แก้ไขรายการอาหารของร้าน
- **ระบบจัดการคอร์สโอมากาเสะ** : ระบบนี้ เป็นระบบจัดการกับแต่ละ คอร์ส เช่น เพิ่ม, ลบ, แก้ไขรายการอาหาร ราคาและจำนวน ที่นั่งในคอร์สโอมากาเสะนั้น
- **ระบบดูรายการทั้งหมดที่ลูกค้าจอง** : ระบบนี้เป็นระบบ ตรวจสอบรายการคอร์สโอมากาเสะ ที่ลูกค้าจองไว้
- **ระบบรับชำระเงินสุทธิ** : ระบบนี้เป็นการชำระเงินของลูกค้า ดำเนินการโดยพนักงาน
- **ระบบจองคอร์สโอมากาเสะ** : ระบบนี้เป็นขั้นตอนที่ลูกค้าเข้ามาทำการจองคอร์สโอมากาเสะ
- **ระบบชำระเงินมัดจำ** : ระบบนี้เป็น ระบบการชำระเงินมัดจำของลูกค้า
- **ระบบยกเลิกการจอง** : ระบบนี้เป็นการยกเลิกการจองคอร์สโอมากาเสะโดยลูกค้า ทำการยกเลิกเอง
- **ระบบตรวจสอบคอร์สโอมากาเสะที่จองไว้** : ระบบนี้เป็นระบบส่วนที่ลูกค้าเข้ามาดูรายละเอียดคอร์สโอมากาเสะของตัวเองที่จองไว้

3. ข้อกำหนดด้านเทคนิค

- **เทคโนโลยีการจัดเก็บข้อมูล**: ใช้เทคโนโลยีการประมวลผลแบบกลุ่มเมฆ (Cloud) เพื่อความปลอดภัยและความเป็นส่วนตัวของข้อมูลผู้ใช้

4. ข้อจำกัดของระบบ

- ความสามารถในการรองรับจำนวนผู้ใช้สูงสุดในช่วงเวลาเดียวกัน

5. ความต้องการด้านความปลอดภัย

- มีมาตรการรักษาความปลอดภัยสำหรับข้อมูลร้านอาหาร รวมถึงการเข้ารหัสข้อมูล การตรวจสอบสิทธิ์การใช้งาน และการป้องกันการเข้าถึงข้อมูลที่ไม่ได้รับอนุญาต

6. ข้อกำหนดด้านการใช้งาน

- **การสมัครสมาชิก** : ผู้ใช้ต้องสมัครสมาชิกก่อนเริ่มใช้งาน หากผู้ใช้ไม่มีสมาชิกจะไม่สารารถเริ่มต้นการใช้งาน ระบบของเว็บได้ และทำได้เพี ยงดูเมนู และรายการอาหาร

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. เสริมสร้างความรู้และทักษะด้าน cloud computing การออกแบบโครงสร้างสถาปัตยกรรม Cloud ที่เหมาะสมเข้ากับโครงงาน เรียนรู้เทคโนโลยีและเครื่องมือที่ใช้ใน Cloud การจัดการความปลอดภัยและความมั่นคง อีกทั้งการทดสอบและการบำรุงรักษาระบบ

2. ได้รับประสบการณ์ในการวางแผน, ติดตามความก้าวหน้า และแก้ไขปัญหา

3. ช่วยลดความซับซ้อนในการจัดการร้าน ทั้งในเรื่องของเมนู การจอง การชำระเงิน และการสื่อสารกับลูกค้า

4. การให้บริการที่เป็นส่วนตัวและสะดวกสบายผ่านเว็ปแอปพลิเคชันจะช่วยเพิ่มความพึงพอใจของลูกค้า

5. สามารถจองโต๊ะ, ดูเมนู และชำระเงินผ่านแอปพลิเคชันได้อย่างง่ายดาย โดยไม่ต้องติดต่อร้านโดยตรง

6. เว็บแอปพลิเคชันสามารถใช้เป็นเครื่องมือทางการตลาดเพื่อดึงดูดลูกค้าใหม่ๆผ่านทางการเสนอโปรโมชั่นหรือการประชาสัมพันธ์ออนไลน์

1.5 โครงสร้างของรายงาน

รายงานฉบับนี้ มีโครงสร้างของรายงานดังต่อไปนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ ขอบเขตของระบบ และประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีที่เกี่ยวข้องและเทคโนโลยีที่ได้นำมาใช้ประกอบการศึกษาและพัฒนาแอปพลิเคชัน

บทที่ 3 การออกแบบระบบ และพัฒนาระบบ รวมถึงการวิเคราะห์ความต้องการของระบบ การออกแบบสถาปัตยกรรมของคลาวด์แอปพลิเคชัน และหลักการทำงานของระบบ

บทที่ 4 ผลการทดลองโดยแบ่งออกเป็นส่วนของผู้ดูแลระบบและผู้ใช้งาน รวมถึงรายละเอียดอื่นๆ

บทที่ 5 สรุปผลการดำเนินงานของโครงงาน โดยสรุปผลที่ได้จากการดำเนินงานทั้งหมด และประเมินผลความสำเร็จของโครงการตามวัตถุประสงค์

บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

1. ทฤษฎีที่เกี่ยวข้อง

1.1 Omakase Restaurant Business

1.1.1 Omakase เป็นคำภาษาญี่ปุ่นที่แปลว่า "ฉันขอฝากคุณจัดการ" หรือ "ตามใจเชฟ" ในบริบทของร้านอาหาร โดยเฉพาะร้านซูชิหรืออาหารญี่ปุ่นแบบดั้งเดิม การสั่งอาหารแบบ Omakase หมายถึงการมอบความไว้วางใจให้กับเชฟในการเลือกสรรและจัดเตรียมเมนูอาหารที่ดีที่สุดตามความสามารถของเชฟและวัตถุดิบที่มีในวันนั้น

1.1.2 การจัดการเมนูและคอร์สอาหาร (Menu and Course Management) ของระบบร้านโอมากะเสะ วัตถุดิบที่ใช้ในร้าน Omakase มักขึ้นอยู่กับฤดูกาล ดังนั้นเชฟจะเปลี่ยนแปลงเมนูตามวัตถุดิบที่ดีที่สุดในแต่ละช่วงเวลา ระบบการจองเป็นสิ่งสำคัญในร้าน Omakase เพราะต้องจัดการจำนวนที่นั่งและการจองล่วงหน้า

1.1.3 การจัดการการจอง (Reservation Management) โดยระบบของร้านโอมากะเสะลูกค้ามักจะมาก่อนเวลาซึ่งบางครั้งก็ไม่สามารถเข้าไปใช้บริการได้จนกว่าลูกค้ารายก่อนหน้าจะใช้บริการเรียบร้อย ซึ่งจะทำให้ในร้าน Omakase ส่วนใหญ่มักจะมีลูกค้ารอคิว และควรที่จะมีระบบจองเพื่อให้มีการจัดการรายชื่อการจองได้อย่างมีประสิทธิภาพ

1.1.4 การจัดการลูกค้า (Customer Management) การยกเลิกการจองที่ชัดเจน เพื่อให้ลูกค้าเข้าใจและปฏิบัติตาอย่างเหมาะสม รวมถึงการเก็บข้อมูลความชอบของลูกค้า เช่น การแพ้อาหาร หรือวัตถุดิบที่ไม่ต้องการ เป็นสิ่งที่สำคัญสำหรับร้าน Omakase ที่เน้นการให้บริการแบบเฉพาะบุคคล

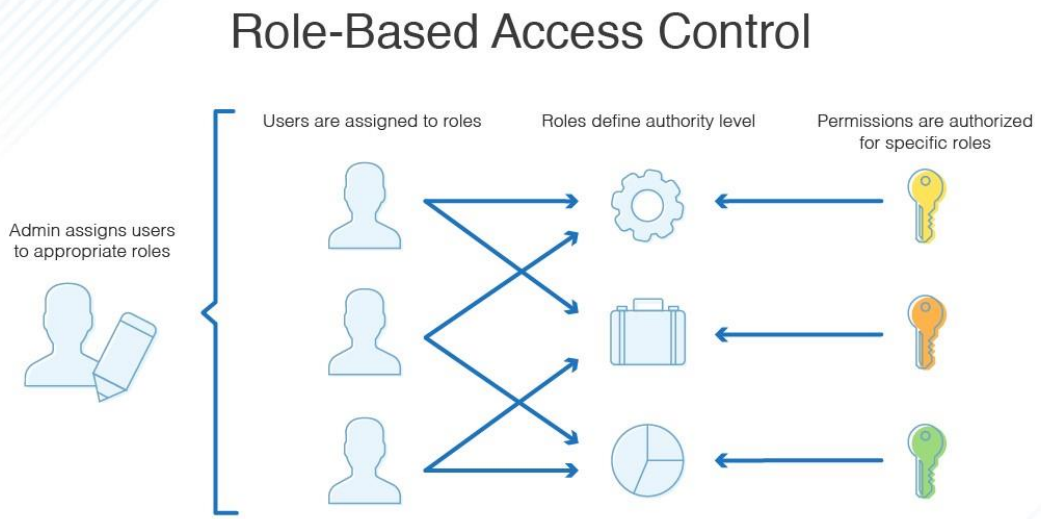
1.2 User Authentication & Authorization

1.2.1 การยืนยันตัวตน (Authentication)เป็นกระบวนการยืนยันว่าผู้ที่กำลังพยายามเข้าใช้งานเป็นตัวจริงของบัญชีนั้นๆ โดยมักจะใช้ชุดข้อมูลยืนยัน เช่น ชื่อผู้ใช้และรหัสผ่าน เป็นต้น

Basic Authentication เป็นการยืนยันตัวตนที่ง่ายที่สุด โดยผู้ใช้จะต้องใส่ชื่อผู้ใช้ (username) และรหัสผ่าน (password) เพื่อเข้าสู่ระบบ

1.2.2 การอนุญาต (Authorization)

Role-Based Access Control (RBAC) เป็นการการควบคุมการเข้าถึงตามบทบาทและคุณลักษณะ ในการเข้าใช้งานในส่วนต่างๆ ของ application ที่เราได้สร้างขึ้น โดยเมื่อมีการเรียกใช้งานในแต่ละส่วน RBAC จะคอยเช็คผู้ใช้งานแต่ละคนว่า มีสิทธิ์ที่จะเข้าใช้งานในส่วนนี้หรือไม่ หากมีก็จะยอมให้เข้าใช้งาน แต่ถ้าหากไม่มีก็จะทำการแสดง error forbidden ออกไป เพื่อให้ผู้ใช้งานทราบว่าไม่สามารถเข้าใช้งานในส่วนนี้ได้ เป็นการกำหนดสิทธิ์การเข้าถึงตามบทบาทของผู้ใช้ เช่น ผู้ดูแลระบบ (Admin) พนักงานร้าน (Employee) และลูกค้า (Customer)



รูปที่ **2.1** ภาพประกอบ **Role-Based Access Control**

1.3 Web Application แอปที่ถูกเขียนขึ้นมาให้สามารถเปิดใช้ใน Web browser ได้โดยตรง ไม่ต้องโหลด Application แบบเต็มๆ ลงเครื่อง ทำให้โดยรวมแล้วกินทรัพยากรค่อนข้างต่ำ สามารถเปิดใช้งานได้ไว ทำหน้าที่คล้ายกับเว็บไซต์ แต่จะสามารถเป็นแอปพลิเคชันได้ด้วย คือเน้นให้ผู้คนเข้ามาใช้งาน มากกว่าดู

โดย Web application จะมีส่วนประกอบการทำงานหลักๆ ที่เห็นกันได้ 4 ส่วน

1. **Web Application** : ตัว Web Application ที่ทำหน้าที่เป็นด่านแรกสุดในการรับข้อมูลจากฝั่งผู้ใช้งาน ซึ่งจะมีการสร้างหรือดัดแปลงการใช้งานไปได้หลากหลายทาง
2. **Web Browser** : เครื่องมือในการเปิด Web Application ซึ่งมีหลากหลายตัวเลือก เช่น Google Chrome Firefox หรือ Microsoft Edge
3. **Web Server** : ระบบ Server ที่ให้บริการแก่บรรดาเว็บไซต์และเว็บแอปพลิเคชั่นต่างๆ ทำหน้าที่รับส่งข้อมูลจากฝั่งผู้ใช้งานและฝั่ง Web Application
4. **Database** : ฐานข้อมูลจากฝั่งผู้ให้บริการ ซึ่งจะทำหน้าที่เก็บข้อมูลที่จำเป็น โดยในบางครั้งมีการทำ Database Server แยกออกมาต่างหาก เพื่อความสะดวกในการควบคุมดูแล และความปลอดภัยของตัว Web Application

รูปที่ **2.2** ภาพประกอบ **Web Application**

1.4 Amazon Web Services (AWS) เป็นบริการบนระบบคลาวด์ที่ครอบคลุมและนำมาใช้มากที่สุดในโลก มีบริการหลากหลายมากกว่า 200 โซลูชัน ตั้งแต่การให้บริการโครงสร้างพื้นฐานด้านไอที การใช้บริการ Server และ Storage การสร้างและดูแลเว็บไซต์ ไปจนถึง ระบบอี-คอมเมิร์ซ การสร้างแอปพลิเคชัน การส่งเสริมการทำงานแบบRemote Working การใช้ระบบ IoT เพื่อการสร้างนวัตกรรมใหม่ ๆ รวมถึงโซลูชันอื่น ๆ อีกมากมาย

รูปที่ **2.3 Amazon Web Services Logo**

2. เทคโนโลยีที่เกี่ยวข้อง

2.1 สถาปัตยกรรม อเมซอน คลาวด์เทคโนโลยี

1. Amazon EC2 (Elastic Compute Cloud)

เป็นบริการที่สามารถรันเซิร์ฟเวอร์เสมือนบนคลาวด์ของ AWS สามารถเลือกขนาดของเซิร์ฟเวอร์ ระบบปฏิบัติการ และการตั้งค่าอื่น ๆ ที่ต้องการ มีการใช้ Security Group เป็น virtual firewall เพื่อควบคุมทราฟฟิกที่เข้าและออกจากเซิร์ฟเวอร์ ช่วยป้องกันการเข้าถึงที่ไม่พึงประสงค์ โดยสามารถกำหนดกฎในการอนุญาตหรือปฏิเสธทราฟฟิกได้ ซึ่งจะเพิ่มความปลอดภัยให้กับเว็บแอปพลิเคชันตามใจเชฟ

เหตุผลที่เลือกใช้เทคโนโลยีนี้ เพราะว่า จะใช้สำหรับรันเซิร์ฟเวอร์หลักที่ให้บริการเว็บแอปพลิเคชันเกี่ยวกับการจัดการฟังก์ชันต่าง ๆ เช่น การจองคอร์สโอมากาเสะ การตรวจสอบการจอง การชำระเงิน การจัดการเมนูและประเภทเมนู การจัดการคอร์ส การดูเมนูคอร์สโอมากาเสะ การดูประวัติการจอง

รูปที่ 2.4 Amazon EC2 Logo

2. **ALB Load Balancer**

Load Balancing คือการกระจายการรับส่งข้อมูลบนเครือข่ายไปยังเซิร์ฟเวอร์หลายตัวที่รองรับแอปพลิเคชัน เพื่อให้สามารถประมวลผลคำขอจากผู้ใช้จำนวนมากได้อย่างรวดเร็วและน่าเชื่อถือ โดย Load Balancer จะทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้และกลุ่มเซิร์ฟเวอร์เพื่อกระจายการใช้งานเซิร์ฟเวอร์อย่างเท่าเทียม

รูปที่ 2.5 ALB Load Balancer

3. Auto Scaling

เป็นบริการที่ช่วยปรับขนาดแอปพลิเคชัน โดยอัตโนมัติเพื่อรักษาประสิทธิภาพและความเสถียร ในราคาที่คุ้มค่า ใช้งานง่ายผ่านอินเทอร์เฟซที่ช่วยให้สร้างแผนการปรับขนาดทรัพยากรต่างๆ เช่น **Amazon EC2, Amazon ECS, DynamoDB** และ **Aurora Replicas** โดยสามารถขยายประสิทธิภาพได้สูงสุดหรือเลือกตัวเลือกที่สมดุลระหว่างประสิทธิภาพและราคา

รูปที่ 2.6 Auto Scaling

4. Cloud 9

AWS Cloud9 คือ **IDE** บนคลาวด์จะช่วยในเรื่องการเขียน รัน และดีบั๊กโค้ดผ่านเบราว์เซอร์ โดยมาพร้อมเครื่องมือสำหรับภาษายอดนิยม เช่น **JavaScript, Python,** และ **PHP** โดยไม่ต้องติดตั้งซอฟต์แวร์ใดๆ ก็สามารถทำงานได้จากทุกที่ที่มีอินเทอร์เน็ต และรองรับการพัฒนาแอปพลิเคชันแบบไร้เซิร์ฟเวอร์ นอกจากนี้ ยังสามารถแชร์สภาพแวดล้อมการพัฒนากับทีมเพื่อทำงานร่วมกันแบบเรียลไทม์

รูปที่ 2.7 Cloud 9

2.2 สถาปัตยกรรมเครือข่าย

1. Amazon VPC

เป็นบริการที่ช่วยให้ผู้จัดทำควบคุมสภาพแวดล้อมระบบเครือข่ายเสมือน ซึ่งรวมถึงการกำหนดทรัพยากร การเชื่อมต่อ และการรักษาความปลอดภัย

เหตุผลที่เลือกใช้เทคโนโลยีนี้ เพราะว่า VPC จะช่วยให้ควบคุมการเชื่อมต่อระหว่างเซิร์ฟเวอร์และฐานข้อมูลได้อย่างปลอดภัย รวมถึงการจัดการความปลอดภัยของข้อมูลที่สำคัญ เช่น ข้อมูลผู้ใช้และข้อมูลการชำระเงิน ทำให้มั่นใจได้ว่าข้อมูลเหล่านี้จะได้รับการปกป้องจากการเข้าถึงที่ไม่ได้รับอนุญาติ

รูปที่ 2.8 Amazon VPC

2.3 สถาปัตยกรรมฐานข้อมูล

1. Amazon RDS (Amazon Relational Database Service) (Database Services)

เป็นบริการฐานข้อมูลเชิงสัมพันธ์ที่ช่วยให้สามารถจัดการฐานข้อมูลที่มีโครงสร้างซับซ้อนได้ รองรับการจัดการการสำรองข้อมูล การปรับขนาด และการจัดการความปลอดภัย

เหตุผลที่เลือกใช้เทคโนโลยีนี้ เพราะว่าเหมาะสำหรับการจัดการข้อมูลที่มีความสัมพันธ์ซับซ้อน เช่น ข้อมูลผู้ใช้, ข้อมูลการชำระเงิน, การสร้างคอร์ส และการจองคอร์ส ซึ่งต้องการการจัดการที่มีความเชื่อถือได้และการเข้าถึงข้อมูลที่รวดเร็ว

รูปที่ 2.9 Amazon Relational Database Service

2.4 เทคโนโลยีที่ใช้

1. **Html (HyperText Markup Language)**

เป็นภาษาหลักที่ใช้ในการสร้างโครงสร้างและเนื้อหาของเว็บเพจ เป็นพื้นฐานสำหรับการแสดงผล

เหตุผลที่เลือกใช้เทคโนโลยีนี้ เพราะว่าเป็นภาษาพื้นฐานที่จำเป็นในการสร้างหน้าเว็บ โครงสร้างของหน้าเว็บ เช่น หน้าสำหรับลงทะเบียน ปุ่ม และข้อความต่าง ๆ

รูปที่ 2.13 HyperText Markup Language (HTML) Logo

2. **CSS (Cascading Style Sheets)**

ใช้สำหรับตกแต่งหน้าเว็ป และจัดรูปแบบ เช่น แบบอักษร, ขนาด, สี และการออกแบบอื่นๆ

เหตุผลที่เลือกใช้เทคโนโลยีนี้ เพราะว่าใช้ในการตกแต่งหน้าเว็บของ Terdinch เพื่อให้มีลักษณะที่น่าสนใจและเข้ากับธีมต่าง ๆ รวมถึงการปรับเปลี่ยนรูปลักษณ์ตามเทศกาล เช่น การปรับเปลี่ยนธีมของขวัญตามฤดูกาลเทศกาล

รูปที่ 2.14 Cascading Style Sheets (CSS) Logo

3. JavaScript

เป็นภาษาโปรแกรมที่ใช้ในการเพิ่มความสามารถในการโต้ตอบกับผู้ใช้ในเว็บเพจ เช่น การตรวจสอบการลงทะเบียน, การสร้างเอฟเฟกต์ต่าง ๆ และการเชื่อมต่อกับ API

เหตุผลที่เลือกใช้เทคโนโลยีนี้ เพราะว่าช่วยให้การเชื่อมต่อและการจัดการบริการต่าง ๆ ของ AWS เป็นไปได้ง่ายและมีประสิทธิภาพ นอกจากนี้ การใช้ JavaScript ยังช่วยลดความซับซ้อนในการจัดการกับระบบ backend และ frontend ทำให้การพัฒนาและปรับปรุงเว็บแอปพลิเคชันเป็นไปอย่างรวดเร็ว และง่ายดาย

รูปที่ 2.15 JavaScript Logo

4. PHP

PHP เป็นภาษาโปรแกรมที่ใช้ในการพัฒนาเว็บแอปพลิเคชันแบบฝั่งเซิร์ฟเวอร์ (Server-side scripting) โดยเฉพาะ ซึ่ง PHP เหมาะสมในการสร้างและจัดการเว็บเพจที่มีการเชื่อมต่อกับฐานข้อมูล, การจัดการกับข้อมูลผู้ใช้, การประมวลผลฟอร์ม, และการแสดงผลเนื้อหาแบบไดนามิก

เหตุผลที่เลือกใช้เทคโนโลยีนี้ เพราะ PHP ทำงานได้ดีเยี่ยมในการเชื่อมต่อและจัดการกับฐานข้อมูลต่าง ๆ ทำให้สามารถพัฒนาเว็บแอปพลิเคชันที่มีความซับซ้อนได้อย่างมีประสิทธิภาพ นอกจากนี้ PHP ยังเป็นภาษาที่มีความยืดหยุ่นและมีการสนับสนุนที่ดีจากชุมชนผู้พัฒนา ทำให้การพัฒนาและปรับปรุงเว็บแอปพลิเคชันเป็นไปได้อย่างรวดเร็วและง่ายดาย

รูปที่ 2.16 PHP Logo

5. Xampp

XAMPP เป็นแพลตฟอร์มการพัฒนาเว็บที่สามารถติดตั้งได้ง่าย ซึ่งรวมเอา Apache HTTP Server, MySQL (หรือ MariaDB), PHP, และ Perl มาไว้ในชุดเดียวกัน XAMPP ถูกออกแบบมาเพื่อให้ผู้พัฒนาสามารถสร้างและทดสอบเว็บแอปพลิเคชันบนเครื่องคอมพิวเตอร์ของตนเองก่อนที่จะทำการเปิดให้ใช้งานจริงบนเซิร์ฟเวอร์

เหตุผลที่เลือกใช้ XAMPP เพราะว่าเป็นเครื่องมือที่มีการติดตั้งและการตั้งค่าที่ง่าย ทำให้ผู้พัฒนาสามารถเริ่มพัฒนาเว็บแอปพลิเคชันได้ทันทีโดยไม่ต้องกังวลเกี่ยวกับการตั้งค่าเซิร์ฟเวอร์หรือการติดตั้งซอฟต์แวร์เสริม นอกจากนี้ XAMPP ยังมีความยืดหยุ่นสูงและสนับสนุนการพัฒนาเว็บแอปพลิเคชันด้วย PHP และ MySQL ทำให้เป็นทางเลือกที่ดีสำหรับการพัฒนา, ทดสอบ, และปรับปรุงเว็บแอปพลิเคชันอย่างมีประสิทธิภาพ

รูปที่ 2.17 XAMPP Logo

6. SQlite

SQLite เป็นระบบจัดการฐานข้อมูลแบบฝังตัว (embedded database) ที่มีขนาดเล็กและเบา โดย SQLite ไม่จำเป็นต้องมีเซิร์ฟเวอร์ในการทำงาน ทำให้เหมาะสำหรับการใช้งานในแอปพลิเคชันที่ต้องการฐานข้อมูลขนาดเล็กถึงขนาดกลาง เช่น แอปพลิเคชันมือถือ, แอปพลิเคชันฝั่งไคลเอนต์, หรือการพัฒนาต้นแบบ (prototyping) ของแอปพลิเคชัน

เหตุผลที่เลือกใช้ SQLite เพราะว่า SQLite ติดตั้งง่ายและมีความรวดเร็วในการดำเนินการฐานข้อมูล อีกทั้งยังเป็นโซลูชันที่ไม่ต้องการการตั้งค่าที่ซับซ้อน ซึ่งช่วยลดเวลาและความยุ่งยากในการ

พัฒนา นอกจากนี้ SQLite ยังรองรับมาตรฐาน SQL และมีการจัดการฐานข้อมูลแบบ self-contained ทำให้การพัฒนาและปรับปรุงแอปพลิเคชันที่ต้องการฐานข้อมูลเป็นไปได้อย่างราบรื่นและมีประสิทธิภาพ

รูปที่ 2.18 SQLite Logo

7. Figma

เป็นเครื่องมือออกแบบอินเทอร์เฟซแบบทำงานร่วมกัน ใช้สำหรับการสร้างแบบจำลอง (Prototype) และพรีเซ้นท์งานเพื่อให้เกิดความเข้าใจที่ตรงกัน และยังสามารถแชร์ Design System และทำให้งานออกแบบในภาพรวมมีความสอดคล้องกัน

เหตุผลที่เลือกใช้เทคโนโลยีนี้ เพราะว่าเหมาะสำหรับการออกแบบเว็ปแอปพลิเคชันและการทำงานร่วมกัน ช่วยให้ผู้พัฒนามองเห็นภาพรวมของเว็ปแอปพลิเคชัน และทำให้สามารถออกแบบและเข้าใจงานได้ง่ายขึ้น

รูปที่ 2.19 Figma Logo

บทที่ 3

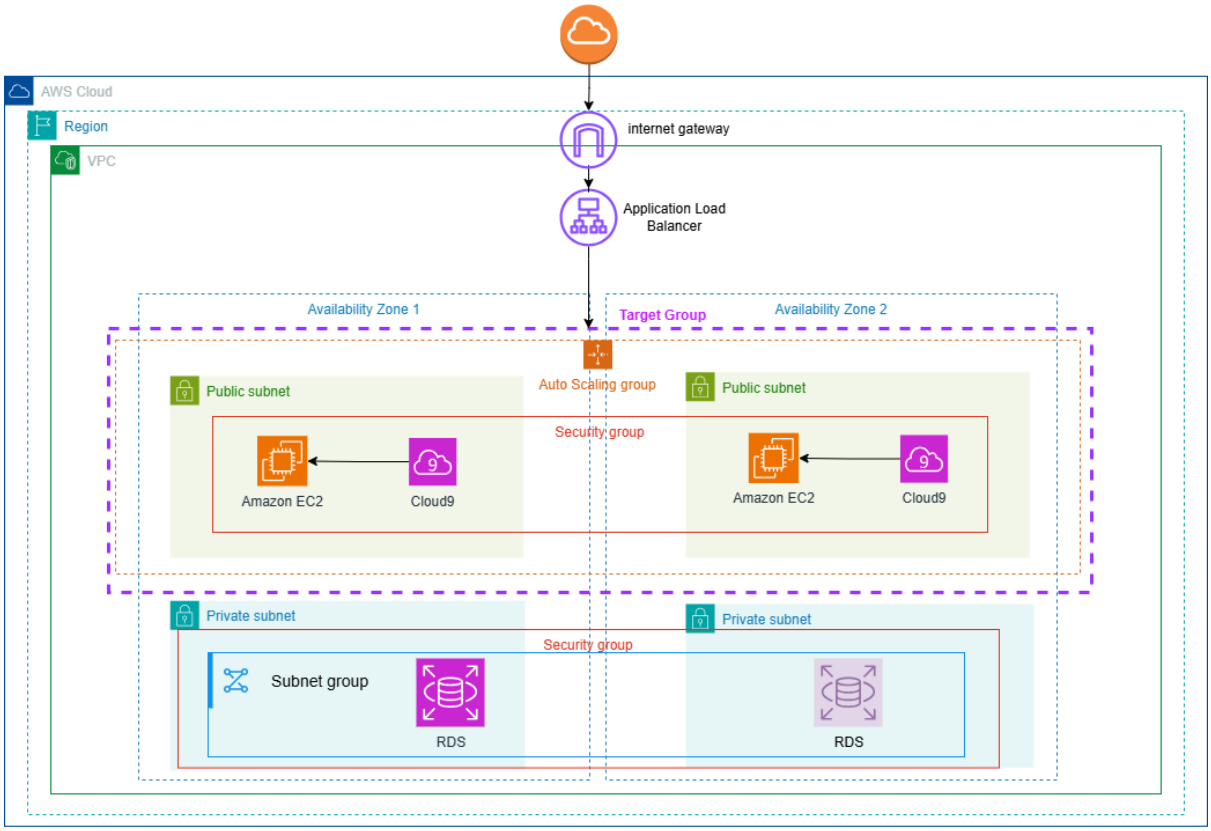
3. ขั้นตอนการดำเนินงานของโครงงาน

3.1 ความต้องการของระบบ

1. ผู้ใช้สามารถสั่ง/เพิ่ม/ลด/แก้ไข รายการอาหารที่สั่งได้
2. ผู้ใช้สามารถใช้งานระบบได้ ด้วยความเสถียร์และมีการอัปเดทข้อมูลอยู่เสมอ
3. ผู้ใช้สามารถใช้งานแอพบนเว็บไซต์ได้อย่างมีประสิทธิภาพ
4. ข้อมูลของร้านอาหารมีความเป็นส่วนตัว และจะเข้าถึงได้ก็ต่อเมื่อได้รับอนุญาติ
5. ผู้ใช้สามารถเข้าใช้งานเว็บได้อย่างรวดเร็ว

3.2 การวิเคราะห์และออกแบบระบบ

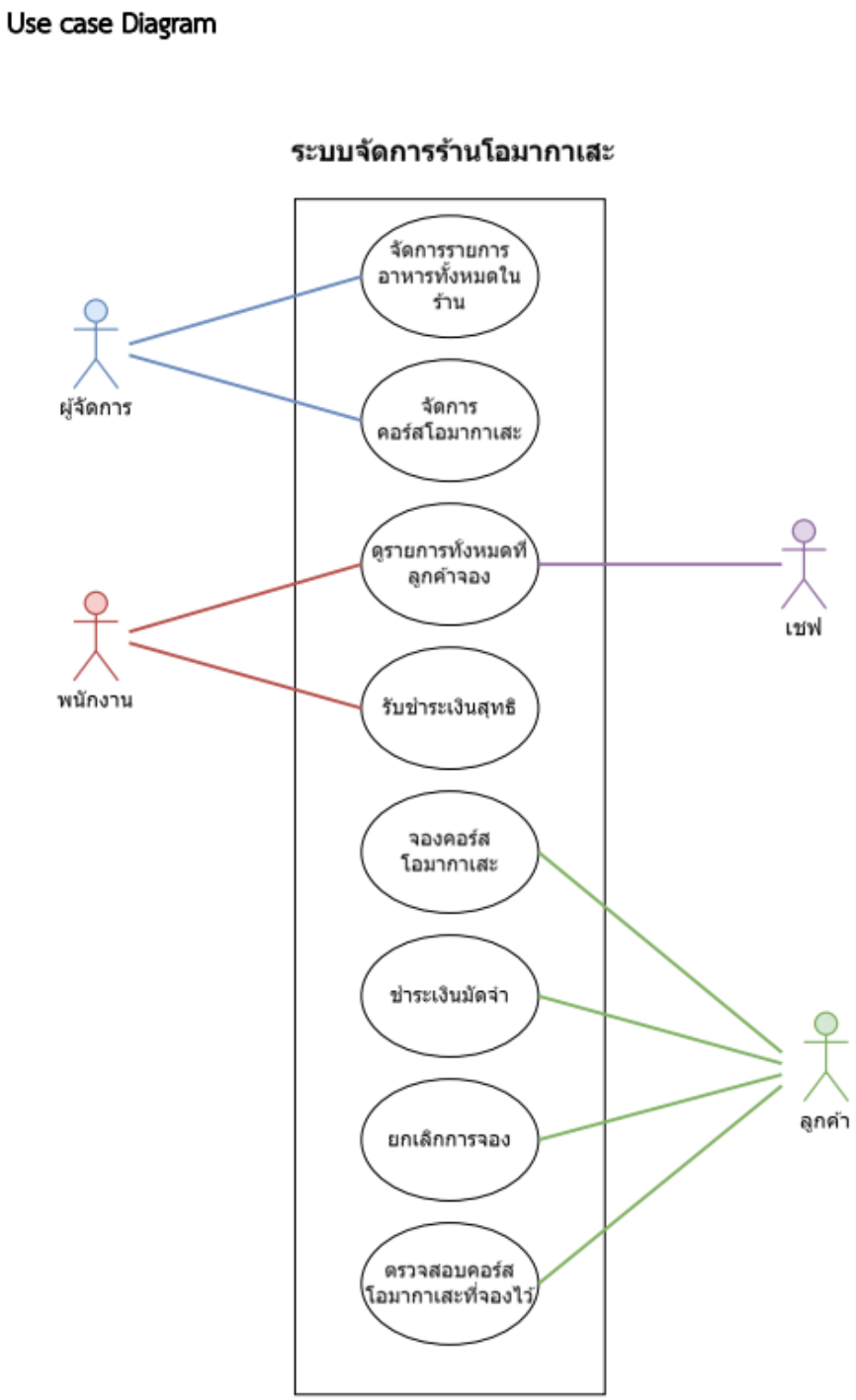
3.2.1. การออกแบบโครงสร้างการใช้งานของ AWS (Amazon Web Service)



รูปที่ 3.1 โครงสร้างการใช้งาน AWS

3.3 หลักการทำงานของระบบ

3.3.1 Use Case Diagram



รูปที่ 3.2 Use Case Diagram

3.3.2 Use Case Description

Use Case Description : จัดการรายการอาหารทั้งหมดในร้าน

Use Case Name: จัดการรายการอาหารทั้งหมดในร้าน	ID: 1
Actor: ผู้จัดการ	
Stakeholders and Interests: เชฟ, ลูกค้า, ผู้จัดการ	
Brief Description: use case นี้เป็นการเพิ่ม ลบ แก้ไขรายการอาหารของร้าน	
Normal Flow of จัดการรายการอาหารทั้งหมดในร้าน:	
ผู้จัดการ	ระบบ
1. ผู้จัดการต้องการจัดการกับรายการอาหารทั้งหมด ของร้าน	
	2. ระบบแสดงหน้าจัดการอาหารของร้าน
3. ผู้จัดการเลือกเพิ่ม ลบ แก้ไขรายการอาหารของร้านตามที่ต้องการ	
Alternate/Exceptional Flows: 3-a : ผู้จัดการเลือกเพิ่มรายการอาหาร 3-a-1 ระบบแสดง ให้เพิ่มรูป ชื่อ ราคาต้นทุนของรายการอาหารนั้น 3-a-2 ผู้จัดการเพิ่มรายละเอียดรายการอาหาร	

3-b : ผู้จัดการเลือกลบรายการอาหาร 3-b-1 ระบบให้ยืนยันการลบอีกครั้ง 3-b-2 ผู้จัดการยืนยันการลบอีกครั้ง 3-c : ผู้จัดการเลือกแก้ไขรายการอาหาร 3-c-1 ระบบแสดงให้แก้ไขรายละเอียดรายการอาหาร 3-c-2 ผู้จัดการแก้ไขรายละเอียดรายการอาหาร

ตารางที่ 3.1 Use Case Description จัดการรายการอาหารทั้งหมดในร้าน

Use Case Description : จัดการคอร์สโอมากาเสะ

Use Case Name: จัดการคอร์สโอมากาเสะ	ID: 2
Actor: ผู้จัดการ	
Stakeholders and Interests: ผู้จัดการ, ลูกค้า, เชฟ	
Brief Description: use case นี้จัดการกับแต่ละ คอร์ส เช่น เพิ่ม, ลบ, แก้ไข รายการอาหาร ราคาและจำนวน ที่นั่งในคอร์สโอมากาเสะนั้น	
Normal Flow of จัดการคอร์สโอมากาเสะ:	
ผู้จัดการ	ระบบ
1. ผู้จัดการต้องการจัดการคอร์สโอมากาเสะ	
	2 . ระบบแสดงหน้าจัดการคอร์สโอมากาเสะ
3. ผู้จัดการเลือกเพิ่ม ลด หรือแก้ไขคอร์สโอมากาเสะ	
4. ผู้จัดการยืนยันการจัดการคอร์สโอมากาเสะ	
	5. ระบบแสดงคอร์สโอมากาเสะ ที่ถูกจัดการแล้ว
Alternate/Exceptional Flows: 3-a : ผู้จัดการเลือกเพิ่มคอร์สโอมากาเสะ	

3-a-1 ระบบขึ้นหน้าให้กรอกข้อมูล เพิ่มช่วงเวลา วันที่ที่เปิดให้บริการ และจำนวนคนที่รองรับ 3-a-2 ผู้จัดการกรอกและบันทึกรายละเอียดของคอร์สโอมากาเสะ 3-b : ผู้จัดการเลือกลบคอร์สโอมากาเสะ 3-b-1 ระบบขึ้นแจ้งเตือนการยืนยันการลบคอร์สโอมากาเสะ 3-b-2 ผู้จัดการยืนยันการลบคอร์สโอมากาเสะ 3-c : ผู้จัดการเลือกแก้ไขคอร์สโอมากาเสะ 3-c-1 ระบบแสดงรายละเอียดให้แก้ไข 3-c-2 ผู้จัดการแก้ไขรายละเอียดของ Omakase Course

ตารางที่ 3.2 Use Case Description จัดการคอร์สโอมากาเสะ

Use Case Description : ดูรายการทั้งหมดที่ลูกค้าจอง

Use Case Name: ดูรายการทั้งหมดที่ลูกค้าจอง	ID: 3
Actor: พนักงาน	
Stakeholders and Interests: พนักงาน, ลูกค้า, เชฟ	
Brief Description: ตรวจสอบรายการคอร์สโอมากาเสะ ที่ลูกค้าจองไว้	
Normal Flow of ดูรายการทั้งหมดที่ลูกค้าจอง:	
พนักงาน	ระบบ
1.พนักงานไปที่รายการอาหาร ดูรายการที่ลูกค้าจอง	
	2. ระบบแสดงรายละเอียดที่ลูกค้าจองทั้งหมด
3. พนักงานค้นหาตามชื่อลูกค้า	
	4. ระบบแสดงชื่อของลูกค้า
5. พนักงานไปที่รายการของลูกค้า	
	6. ระบบแสดงรายละเอียดของลูกค้าของคนๆนั้น

ตารางที่ 3.3 Use Case Description ดูรายการทั้งหมดที่ลูกค้าจอง

Use Case Description : รับชำระเงินสุทธิ

Use Case Name: รับชำระเงินสุทธิ		ID: 4
Actor: พนักงาน		
Stakeholders and Interests: ลูกค้า , พนักงาน, ผู้จัดการ		
Brief Description: use case นี้เป็นการชำระเงินของลูกค้าดำเนินการโดยพนักงาน		
Normal Flow of รับชำระเงินสุทธิ:		
พนักงาน	ระบบ	ลูกค้า
1. 1. พนักงานต้องการคิดค่าอาหารสุทธิ		
	2.ระบบแสดงค่าใช้จ่ายสุทธิของลูกค้า	
3.พนักงานสอบถามช่องทางที่ลูกค้าต้องการชำระเงิน		
		4. ลูกค้าเลือกช่องทางการชำระเงิน
		5. ลูกค้าชำระเงิน

	6. ระบบตรวจสอบการชำระเงิน	
	7. ระบบดำเนินการยืนยันการชำระเงิน	
	8. ระบบดำเนินการส่งการยืนยันการชำระเงินไปให้ลูกค้า	
	9. ระบบบันทึกสถานะเป็นเสร็จสิ้น	
Alternate/Exceptional Flows: 4-a: ลูกค้าเลือกการชำระเงินเป็นการโอน 4-a-1:พนักงานเลือกการชำระเงินเป็นการโอนและแสดง QRcode ให้กับลูกค้า 6-a:ลูกค้าชำระเงินไม่สำเร็จ 6-a-1:ระบบให้ทำการชำระเงินอีกครั้ง		

ตารางที่ 3.4 Use Case Description รับชำระเงินสุทธิ

Use Case Description : จองคอร์สโอมากาเสะ

Use Case Name: จองคอร์สโอมากาเสะ	ID: 5
Actor: ลูกค้า	
Stakeholders and Interests: เชฟ, พนักงาน	
Brief Description: use case นี้ เป็นขั้นตอนที่ลูกค้าเข้ามาทำการจองคอร์สโอมากาเสะ	
Normal Flow of จองคอร์สโอมากาเสะ:	
ลูกค้า	ระบบ
1. ลูกค้าต้องการดูคอร์สโอมากาเสะ	
	2. ระบบแสดงรายการคอร์สโอมากาเสะ
3. ลูกค้าเลือกคอร์สโอมากาเสะที่ต้องการ	
4. ลูกค้าระบุรายละเอียดที่จำเป็น	
5. ลูกค้าชำระเงินมัดจำ	
	6. ระบบดำเนินการจอง

ตารางที่ 3.5 Use Case Description จองคอร์สโอมากาเสะ

Use Case Description : ชำระเงินมัดจำ

Use Case Name: ชำระเงินมัดจำ	ID: 6
Actor: ลูกค้า	
Stakeholders and Interests: ลูกค้า, พนักงาน, ผู้จัดการ	
Brief Description: use case นี้เป็นการชำระเงินมัดจำของลูกค้า	
Normal Flow of ชำระเงินมัดจำ:	
ลูกค้า	ระบบ
1. ลูกค้าต้องการชำระเงินมัดจำ	
	2. ระบบแสดงวิธีการชำระเงินมัดจำและแสดงช่องทางการชำระเงิน
3. ลูกค้าชำระเงินมัดจำ	
	4. ระบบดำเนินการตรวจสอบ
	5. ระบบดำเนินการยืนยันการชำระเงิน
Alternate/Exceptional Flows: 4-a : ระบบตรวจสอบว่าลูกค้าโอนเงินไม่สำเร็จ 4-a-1: ระบบแจ้งให้ลูกค้าดำเนินการโอนเงินมัดจำอีกครั้ง	

ตารางที่ 3.6 Use Case Description ชำระเงินมัดจำ

Use Case Description : ยกเลิกการจอง

Use Case Name: ยกเลิกการจอง	ID: 7
Actor: ลูกค้า	
Stakeholders and Interests: ลูกค้า	
Brief Description: use case นี้ เป็นการยกเลิกการจองคอร์สโอมากาเสะโดยลูกค้าเอง	
Normal Flow of ยกเลิกการจอง:	
ลูกค้า	ระบบ
1. ลูกค้าต้องการยกเลิกการจองคอร์สโอมากาเสะ	
	2. ระบบแสดงรายละเอียดเกี่ยวกับการยกเลิกการจอง
3. ลูกค้ายกเลิกการจอง	
	4. ระบบดำเนินการยกเลิกการจอง

ตารางที่ 3.7 Use Case Description ยกเลิกการจอง

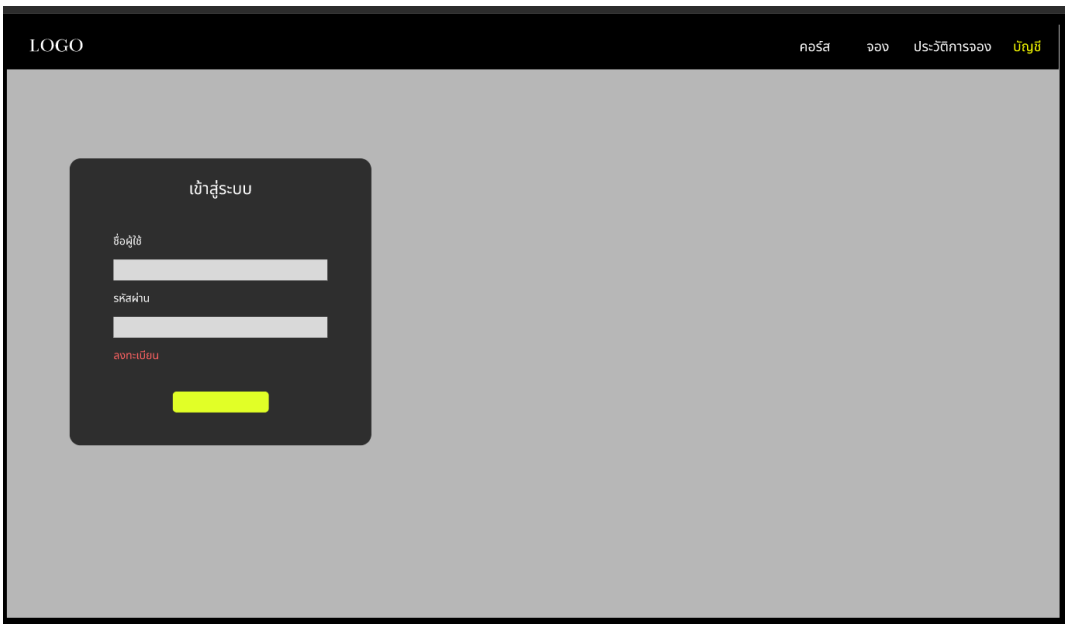
Use Case Description : ตรวจสอบคอร์สโอมากาเสะที่จองไว้

Use Case Name: ตรวจสอบคอร์สโอมากาเสะที่จองไว้	ID: 8
Actor: ลูกค้า	
Stakeholders and Interests: ลูกค้า	
Brief Description: use case นี้ เป็นส่วนที่ลูกค้าเข้ามาดูรายละเอียดคอร์สโอมากาเสะของตัวเองที่จองไว้	
Normal Flow of ตรวจสอบคอร์สโอมากาเสะที่จองไว้:	
ลูกค้า	ระบบ
1. ลูกค้าต้องการดูรายละเอียดคอร์สโอมากาเสะของตัวเองที่จองไว้	
	2. ระบบแสดงแต่ละคอร์สโอมากาเสะของลูกค้าที่จองไว้
3. ลูกค้าเลือกคอร์สโอมากาเสะที่ต้องการดู	
	4. ระบบแสดงคอร์สโอมากาเสะที่ต้องการดู

ตารางที่ 3.8 Use Case Description ตรวจสอบคอร์สโอมากาเสะที่จองไว้

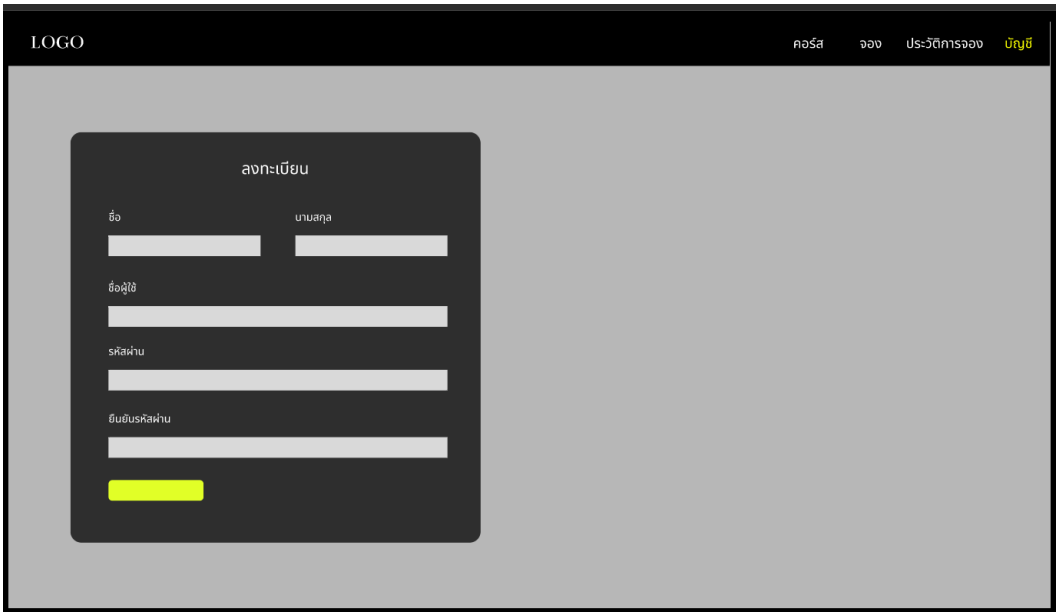
3.3.3 User Interface (Prototype)

หน้าเข้าสู่ระบบ



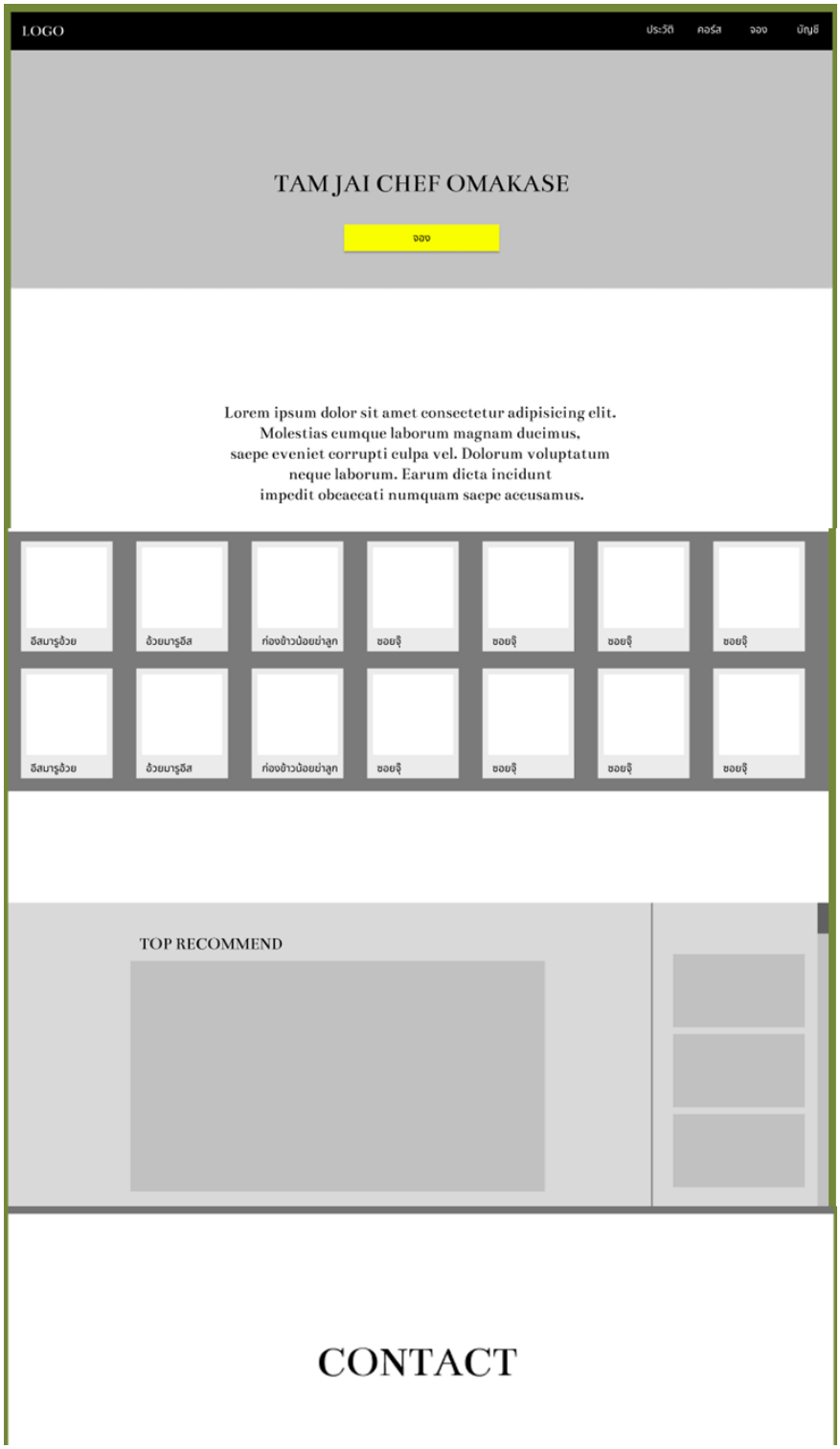
รูปที่ 3.3 เข้าสู่ระบบ

หน้าลงทะเบียน



รูปที่ 3.4 ลงทะเบียน

หน้าหลักของระบบ

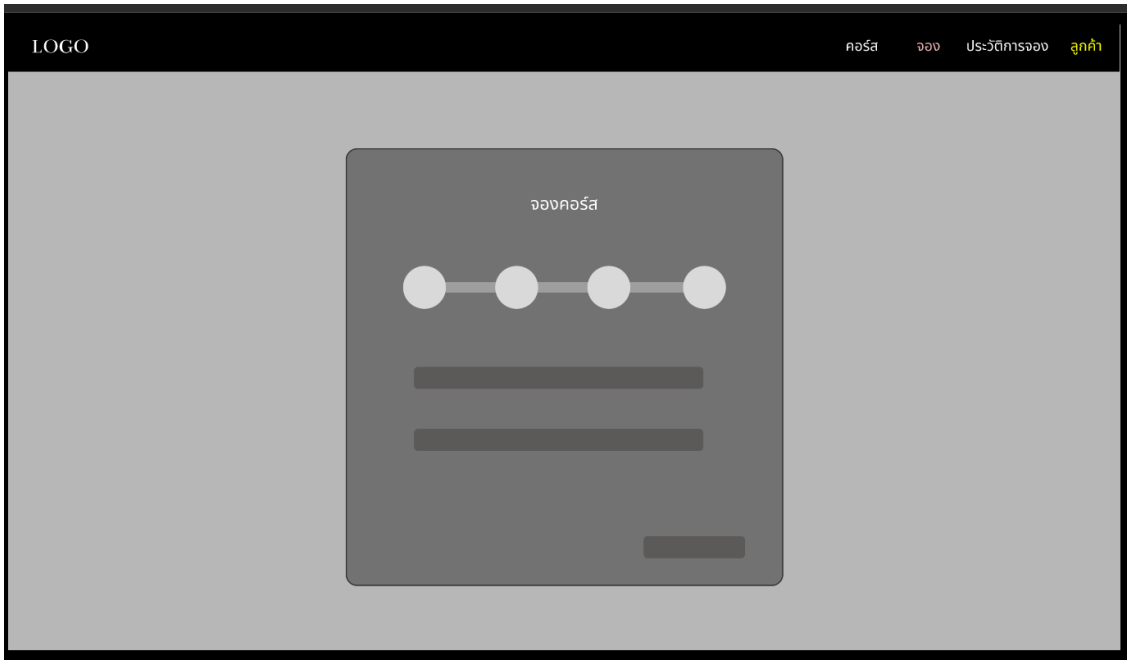


รูปที่ 3.5 หน้าหลักระบบ

หน้าดูรายการเมนูอาหาร

รูปที่ 3.6 ดูรายการเมนูอาหาร

หน้าจองคอร์สโอมากาเสะ

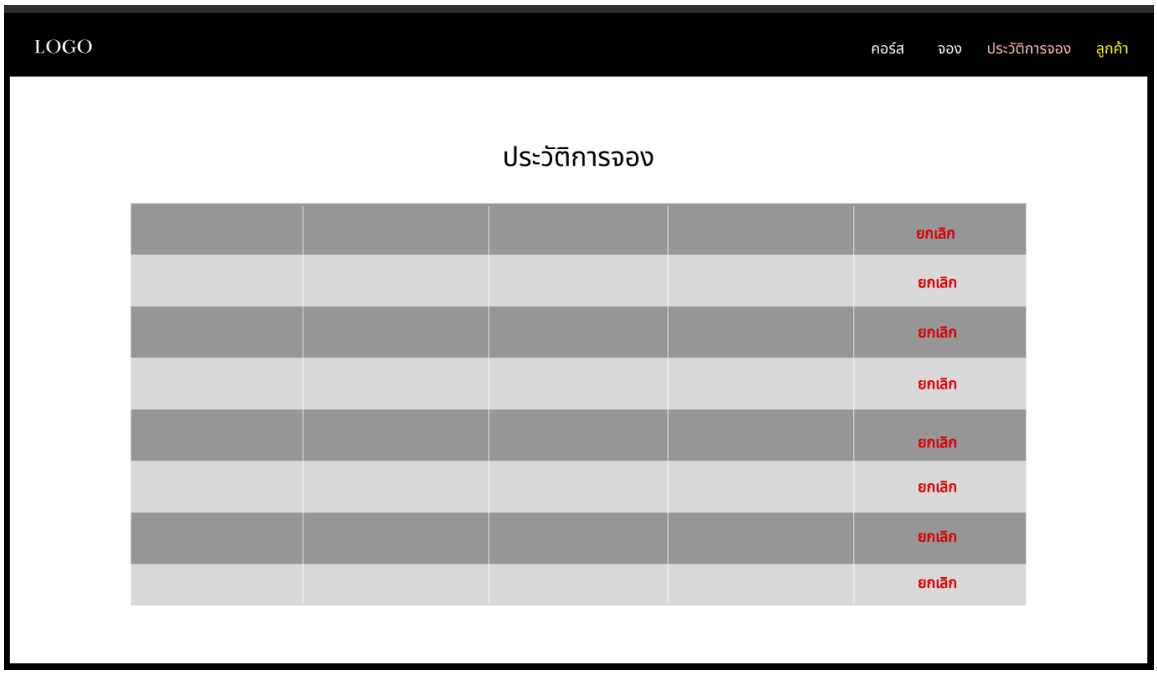


รูปที่ 3.7 จองคอร์สโอมากาเสะ

หน้าชำระเงินมัดจำ

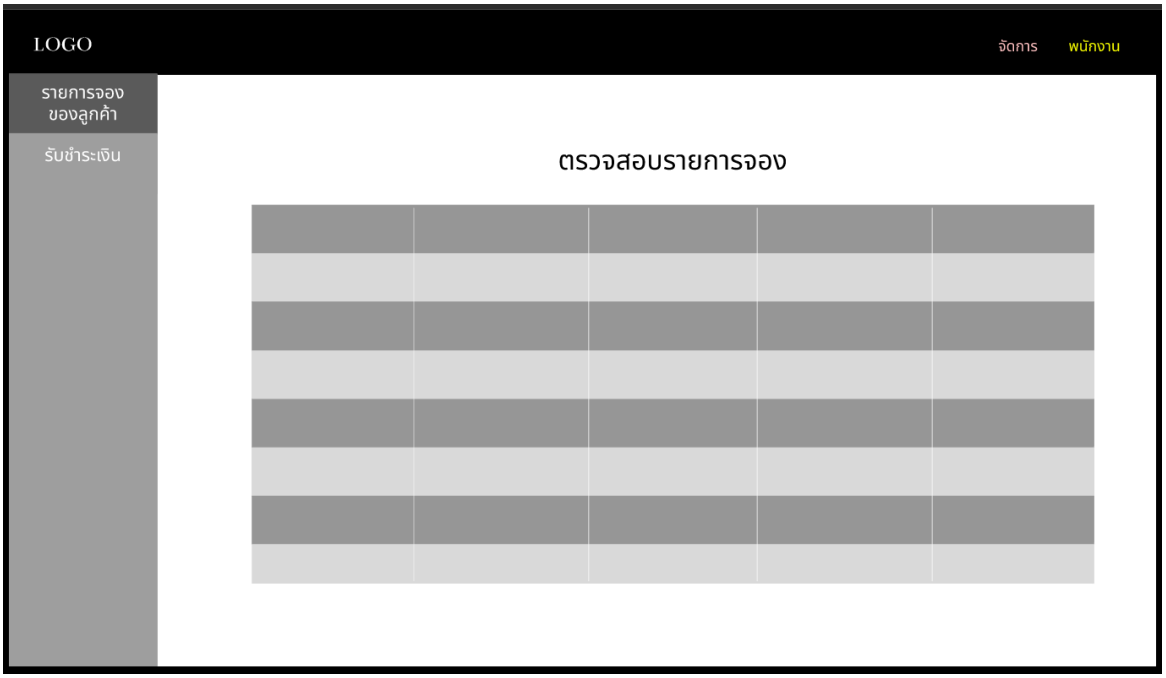
รูปที่ 3.8 ชำระเงินมัดจำ

หน้าตรวจสอบการจองคอร์ส และ ยกเลิกการจอง



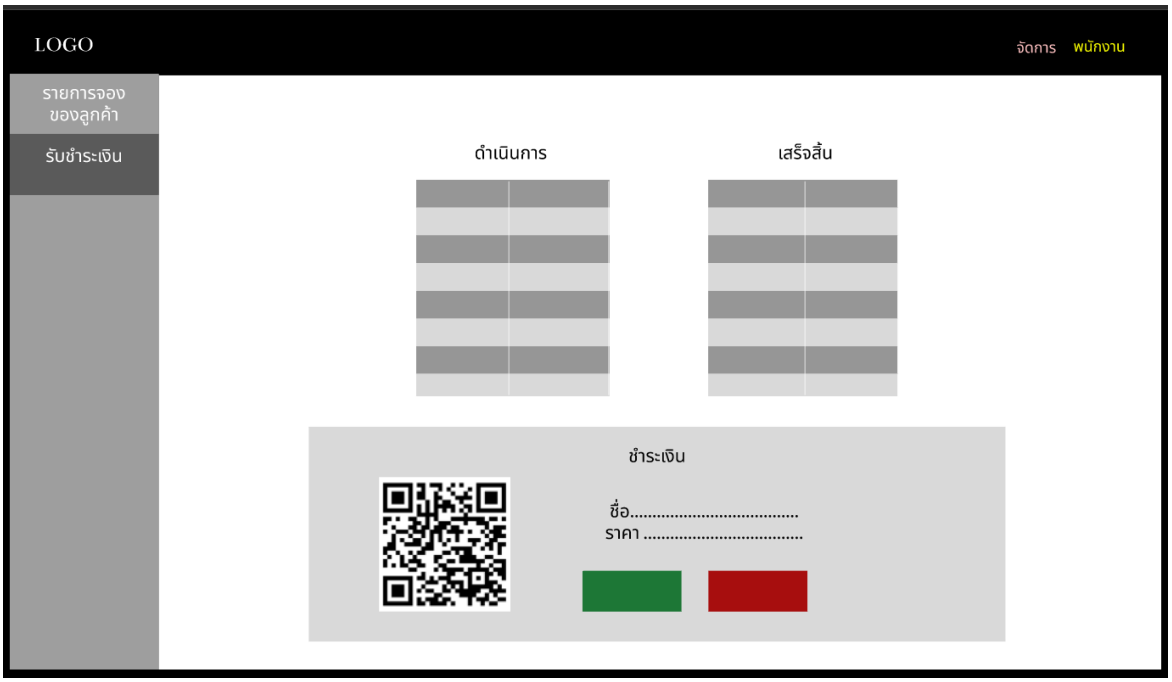
รูปที่ 3.9 ตรวจสอบการจองคอร์ส และ ยกเลิกการจอง

หน้าดูรายการการสั่งอาหาร



รูปที่ 3.10 ดูรายการการสั่งอาหาร

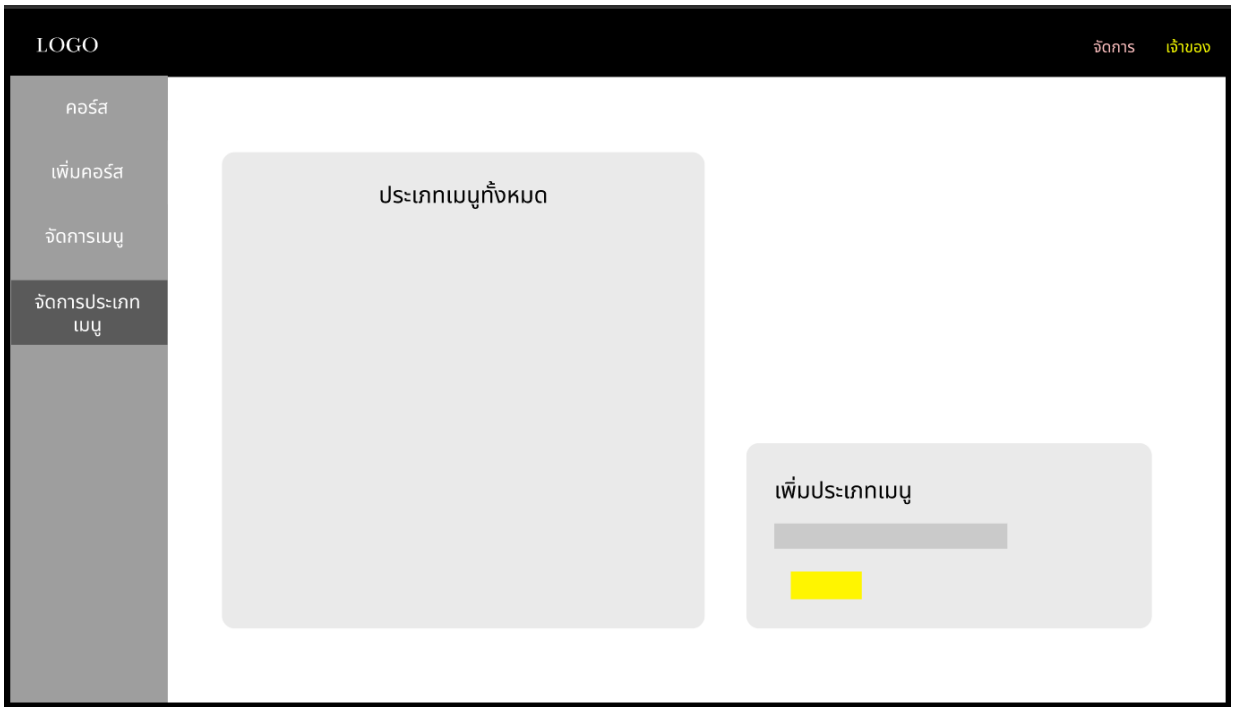
หน้าชำระเงิน



รูปที่ 3.11 ชำระเงิน

หน้าแก้ไข เพิ่ม ลด เมนูอาหารประเภทเมนู

รูปที่ 3.12 แก้ไข เพิ่ม ลด เมนูอาหาร



รูปที่ 3.13 แก้ไข เพิ่ม ลดประเภทเมนู

หน้าเพิ่มคอร์สโอมากาเสะ

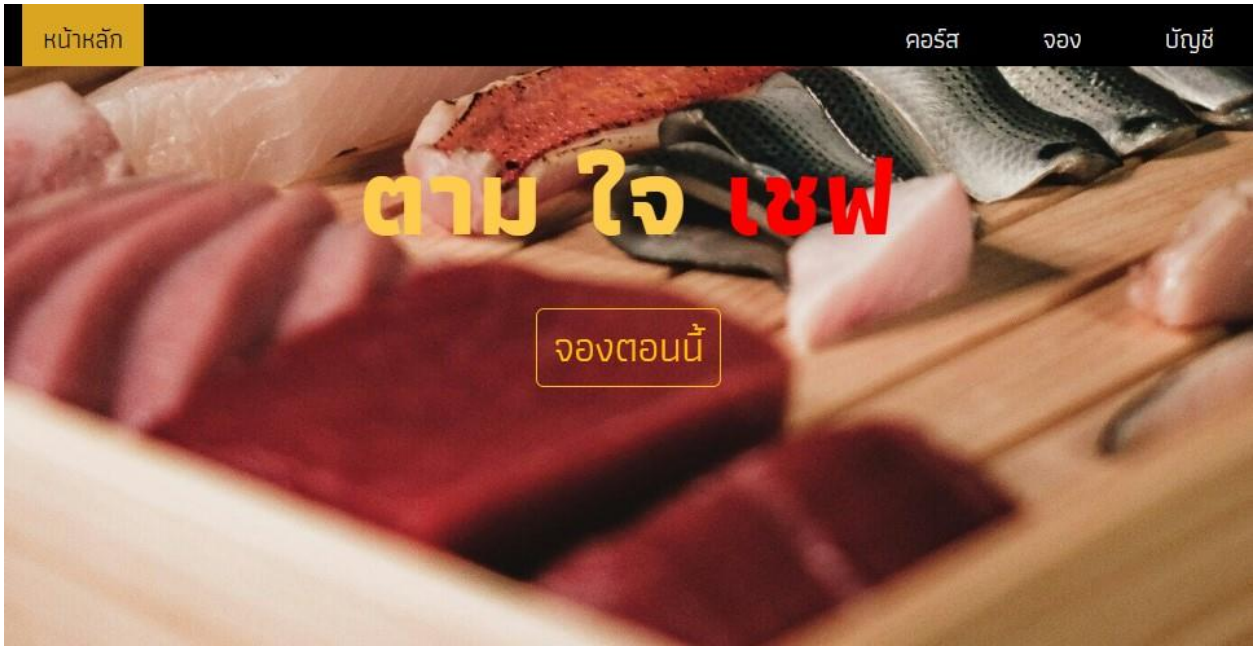
รูปที่ 3.14 เพิ่มคอร์สโอมากาเสะ

บทที่ 4

ระบบต้นแบบ

4.1 ส่วนติดต่อผู้ใช้งาน

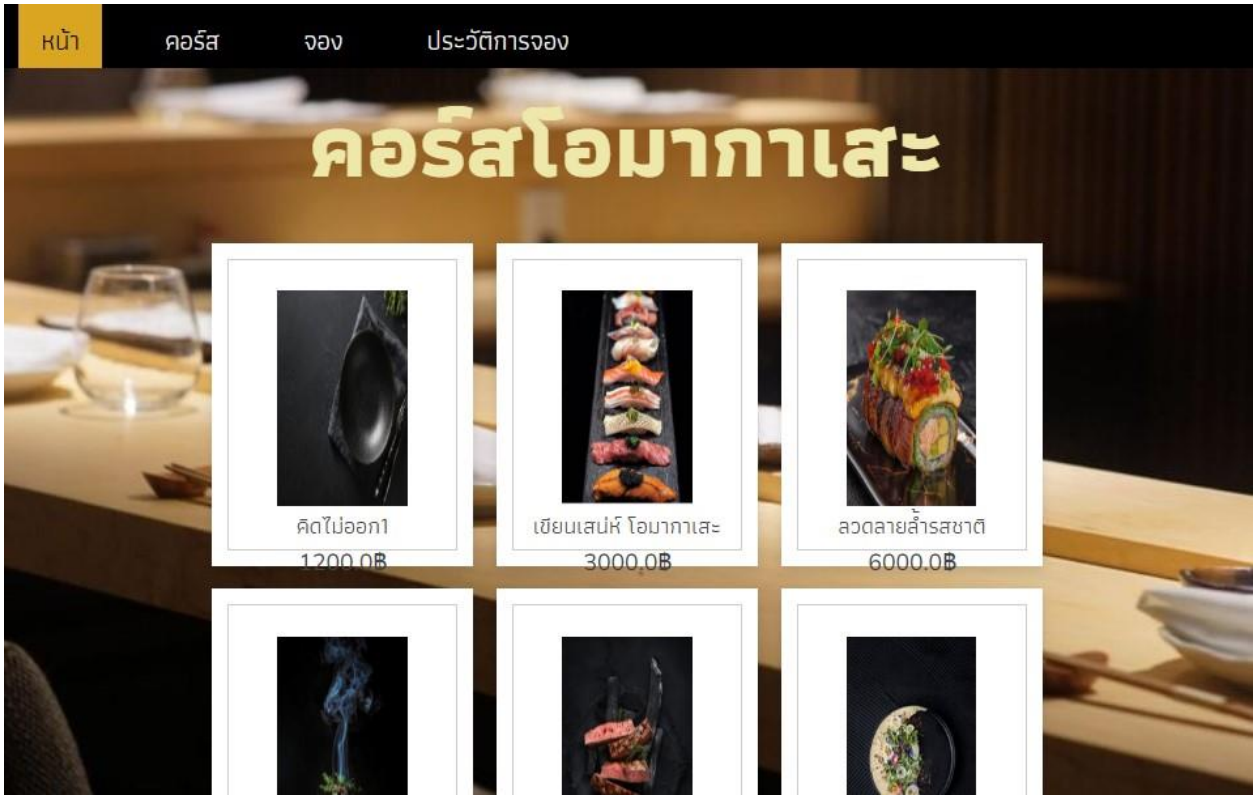
1.หน้าหลัก



รูปที่ 4.1 หน้าหลัก

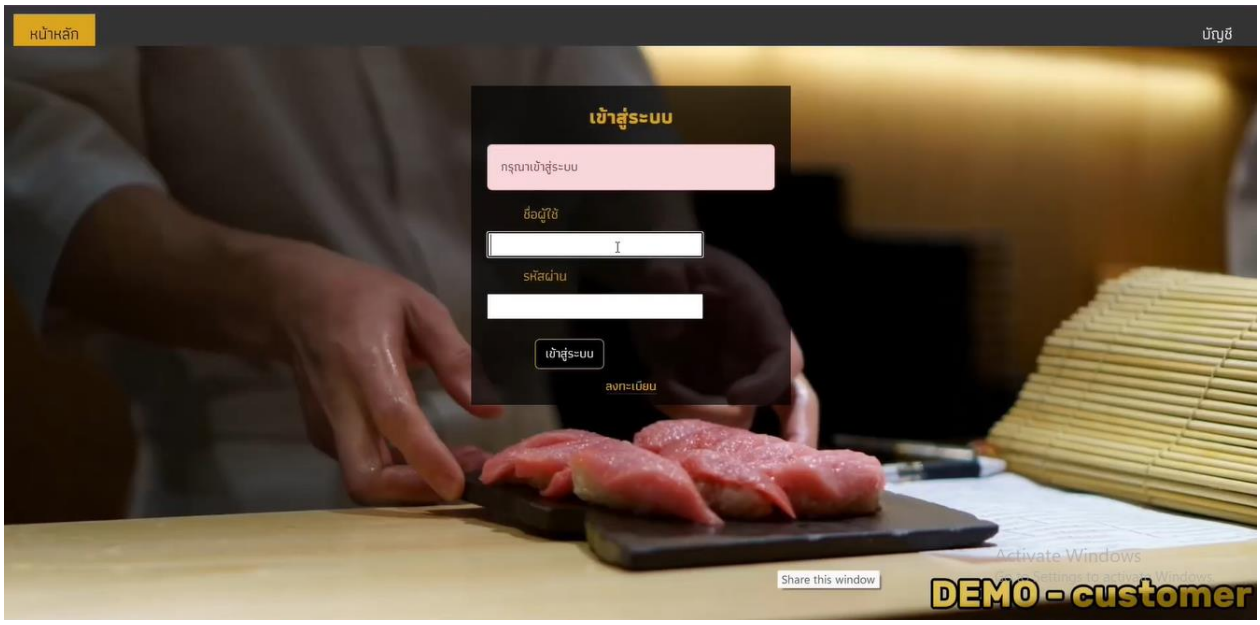
รูปที่ 4.2 หน้าหลัก

2. หน้าดูคอร์สอาหาร



รูปที่ 4.3 หน้าดูคอร์สอาหาร

3.หน้า Log-in

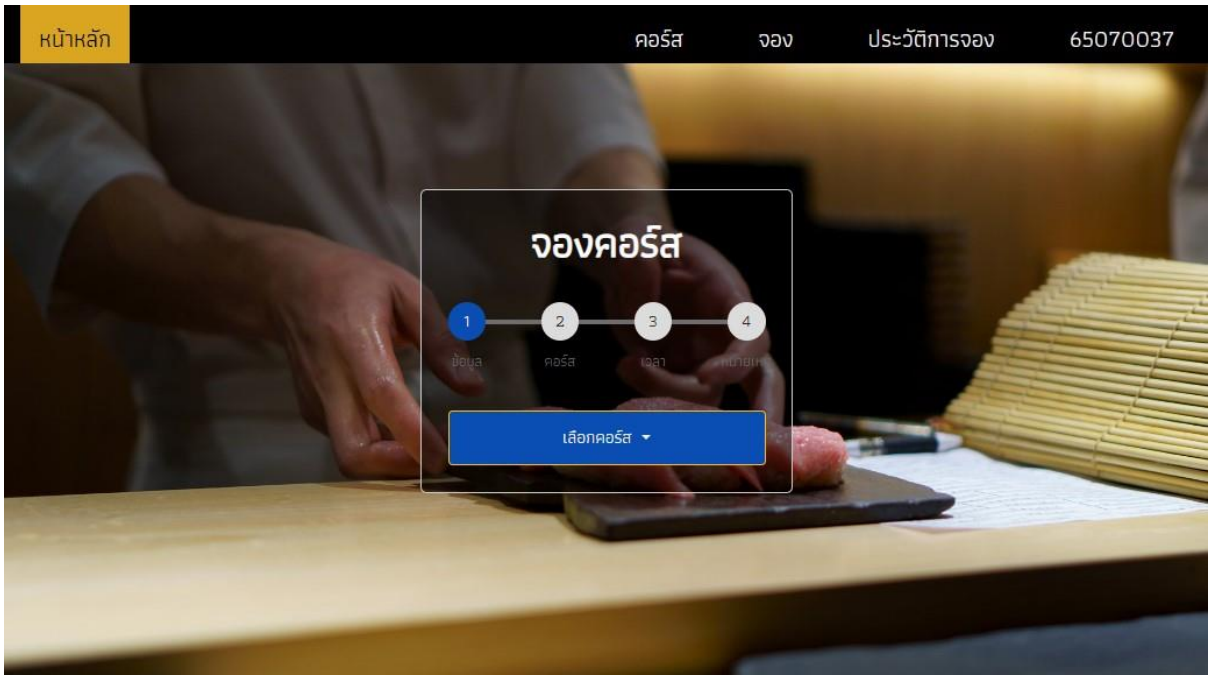


รูปที่ 4.4 หน้า Log-In

4.หน้าหลักลูกค้า

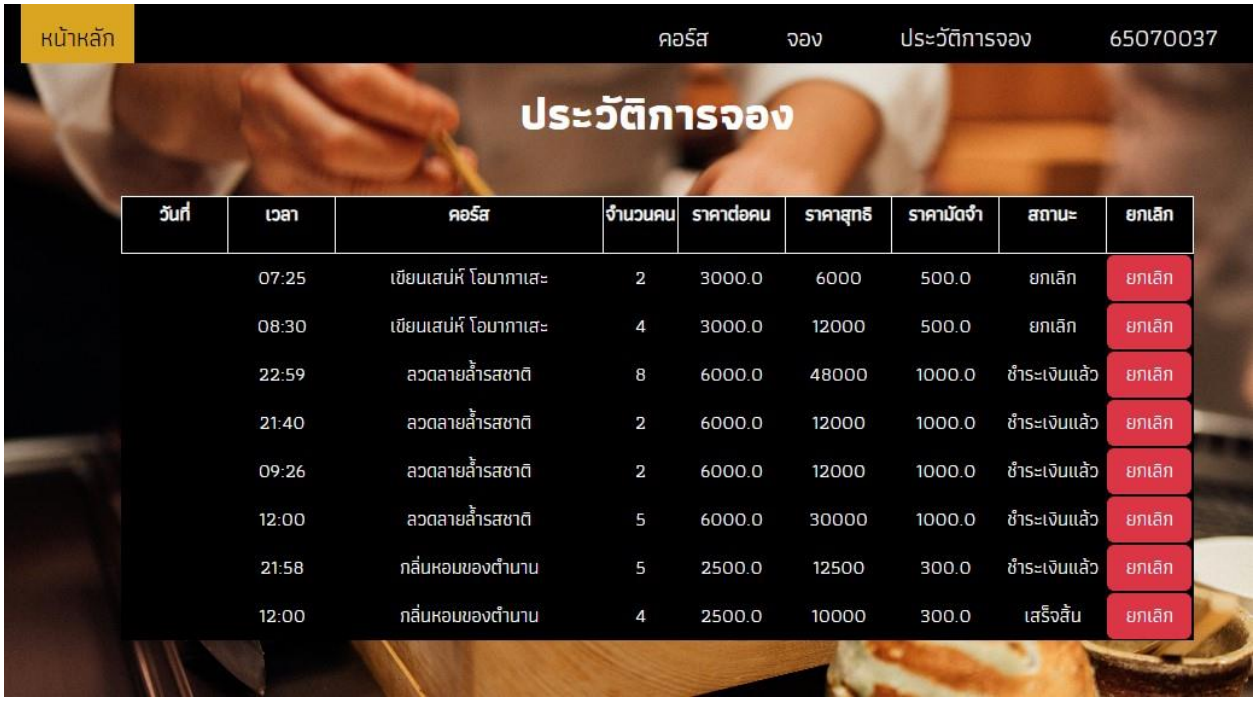
รูปที่ 4.5 หน้าหลักลูกค้า

5.หน้าจองคอร์สโอมากาเสะ (ลูกค้า)



รูปที่ 4.6 หน้าจองคอร์สโอมากาเสะ(ลูกค้า)

6.หน้าดูประวัติการจอง (ลูกค้า)



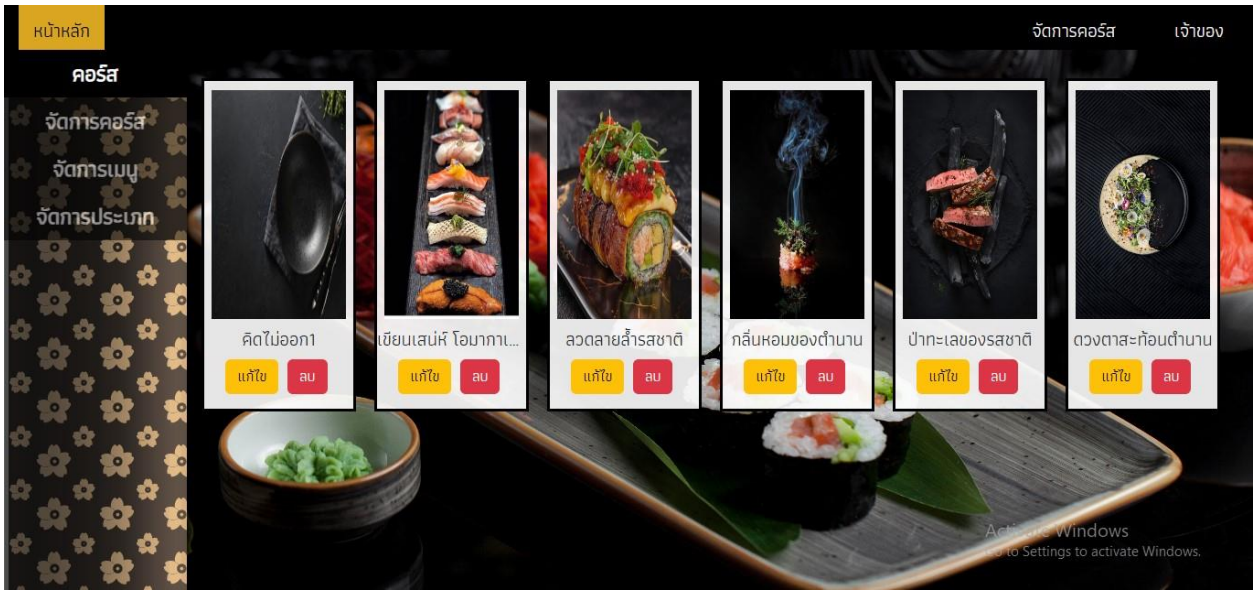
วันที่	เวลา	คอร์ส	จำนวนคน	ราคาต่อคน	ราคาสุทธิ	ราคามัดจำ	สถานะ	ยกเลิก
	07:25	เขียนเสน่ห์ โอมากาเสะ	2	3000.0	6000	500.0	ยกเลิก	ยกเลิก
	08:30	เขียนเสน่ห์ โอมากาเสะ	4	3000.0	12000	500.0	ยกเลิก	ยกเลิก
	22:59	ลวดลายล้ำรสชาติ	8	6000.0	48000	1000.0	ชำระเงินแล้ว	ยกเลิก
	21:40	ลวดลายล้ำรสชาติ	2	6000.0	12000	1000.0	ชำระเงินแล้ว	ยกเลิก
	09:26	ลวดลายล้ำรสชาติ	2	6000.0	12000	1000.0	ชำระเงินแล้ว	ยกเลิก
	12:00	ลวดลายล้ำรสชาติ	5	6000.0	30000	1000.0	ชำระเงินแล้ว	ยกเลิก
	21:58	กลิ่นหอมของตำนาน	5	2500.0	12500	300.0	ชำระเงินแล้ว	ยกเลิก
	12:00	กลิ่นหอมของตำนาน	4	2500.0	10000	300.0	เสร็จสิ้น	ยกเลิก

รูปที่ 4.7 หน้าดูประวัติการจอง (ลูกค้า)

7. หน้าหลักผู้จัดการ

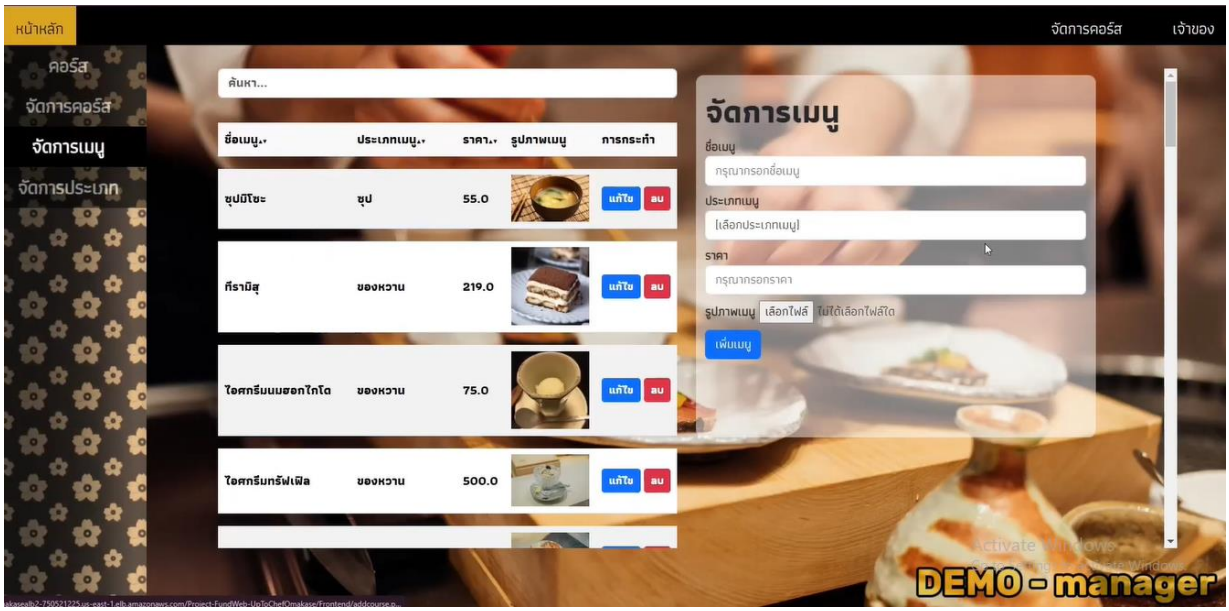
รูปที่ 4.8 หน้าหลักผู้จัดการ

8. หน้าจัดการคอร์ส (ผู้จัดการ)



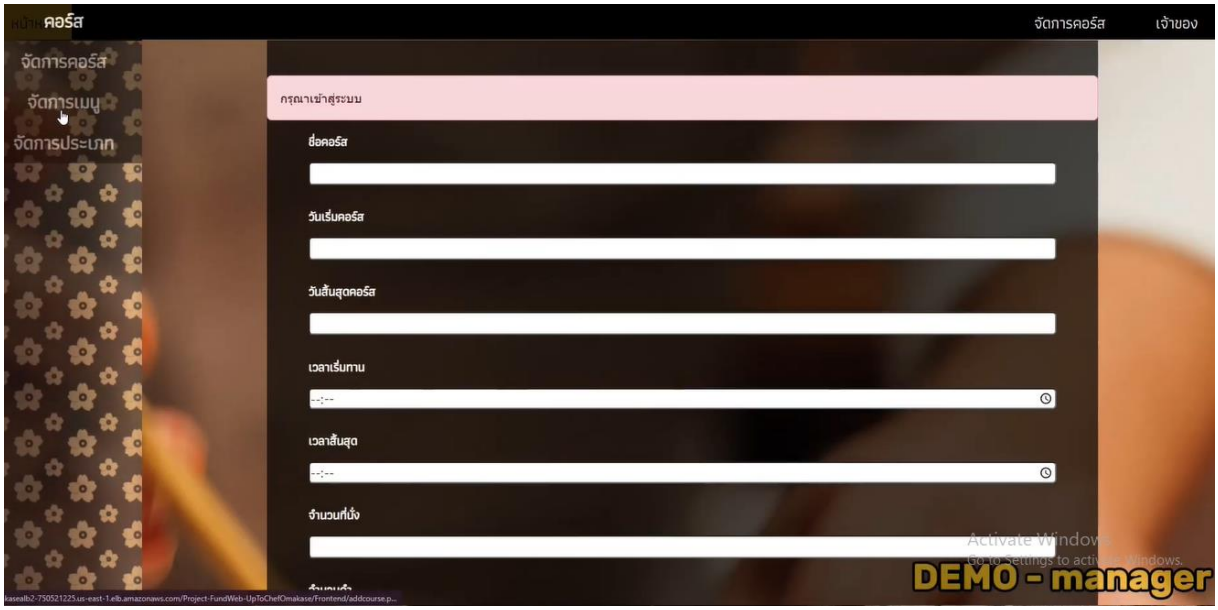
รูปที่ 4.9 หน้าจัดการคอร์ส (ผู้จัดการ)

9. หน้าจัดการเมนู (ผู้จัดการ)

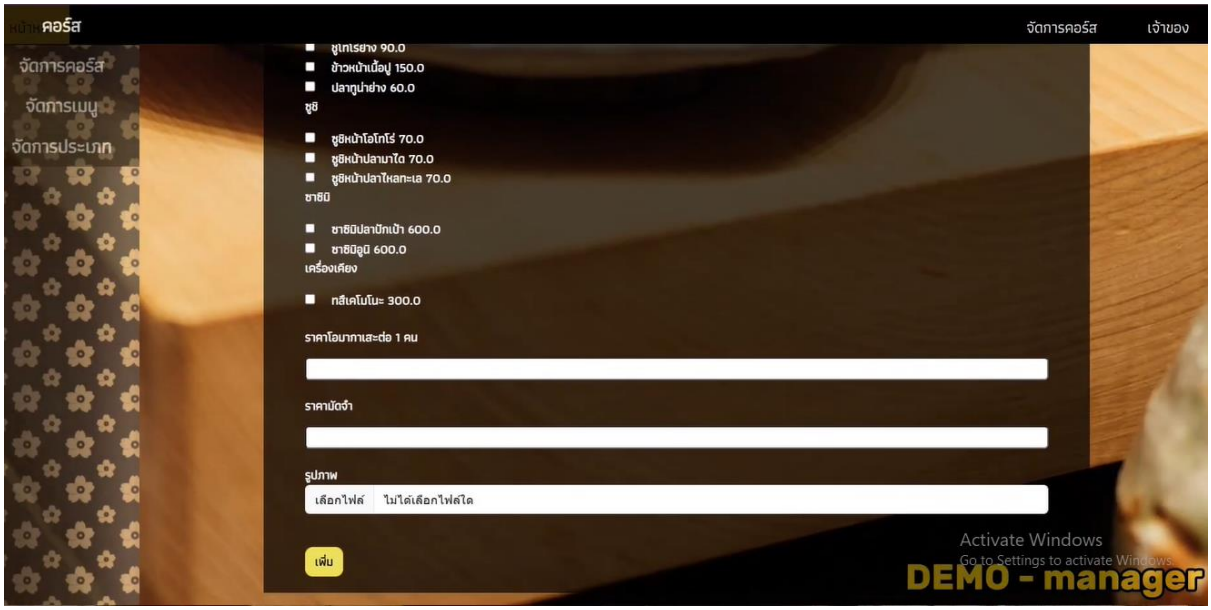


รูปที่ 4.10 หน้าจัดเมนู (ผู้จัดการ)

10. หน้าเพิ่มคอร์ส (ผู้จัดการ)



รูปที่ 4.11 หน้าเพิ่มคอร์ส (ผู้จัดการ)

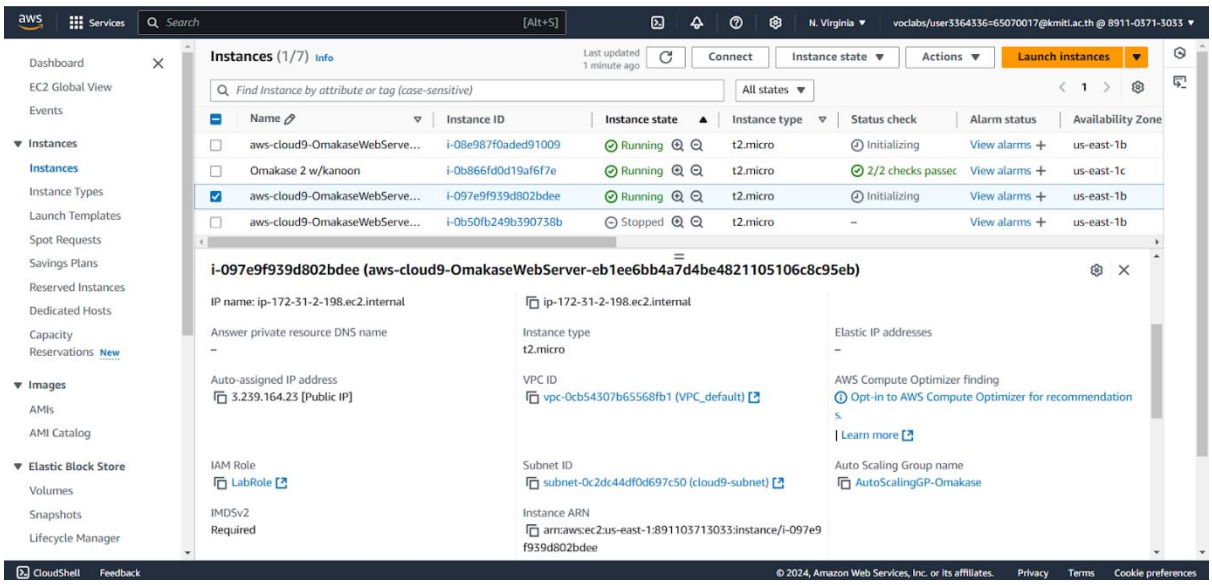


รูปที่ 4.12 หน้าเพิ่มคอร์ส (ผู้จัดการ)

4.2 ผลการดำเนินงานของการใช้งาน AWS ร่วมกับเว็บแอปพลิเคชัน

1. Amazon Elastic Compute Cloud (Amazon EC2)

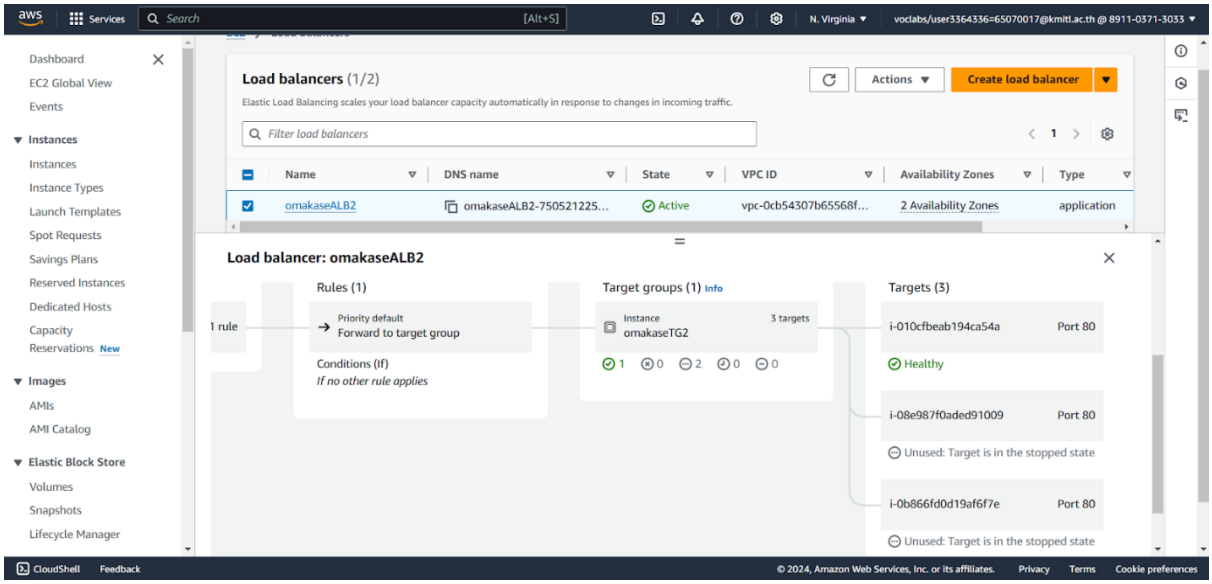
ใช้เป็น Server สำหรับลองรับการใช้งานต่างๆ โดยตัว Instance จะมีการเพิ่มขึ้นหรือลดตามปริมาณการใช้งาน



รูปที่ 4.13 Amazon Elastic Compute Cloud (Amazon EC2)

2. Elastic Load Balancing

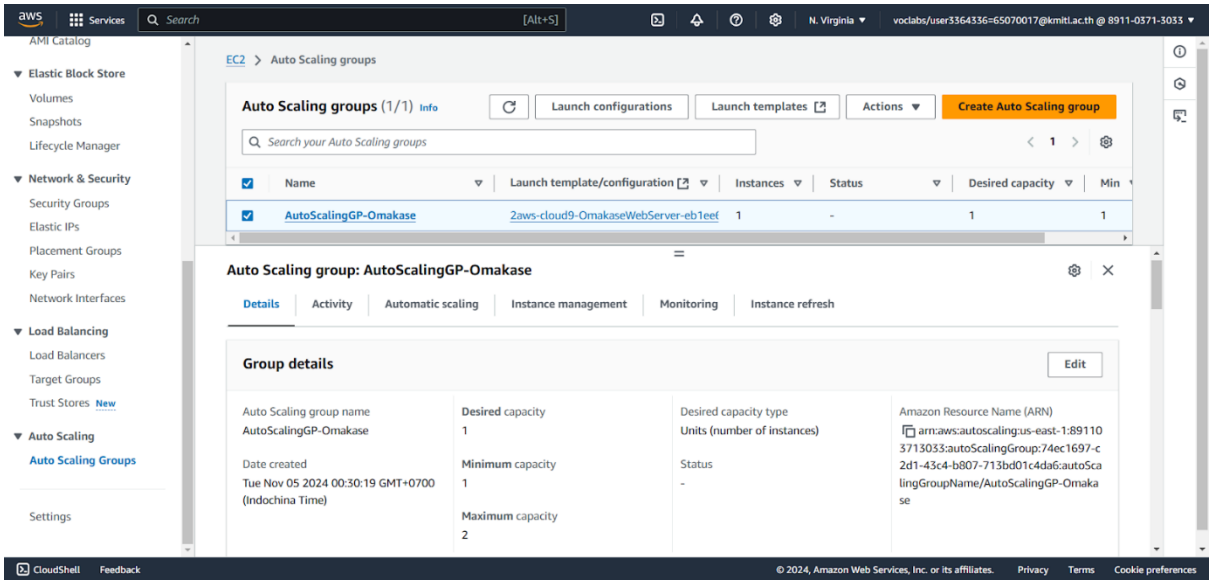
ใช้เพื่อกระจาย load ไปยัง EC2 Instance ที่มี Status เป็น Healthy เพื่อช่วยลดภาระการทำงานของ Server ทำให้ Server ทำงานได้อย่างลื่นไหลและลดโอกาสการเกิดเว็บล่ม โดยเลือกใช้ Application Load Balancer



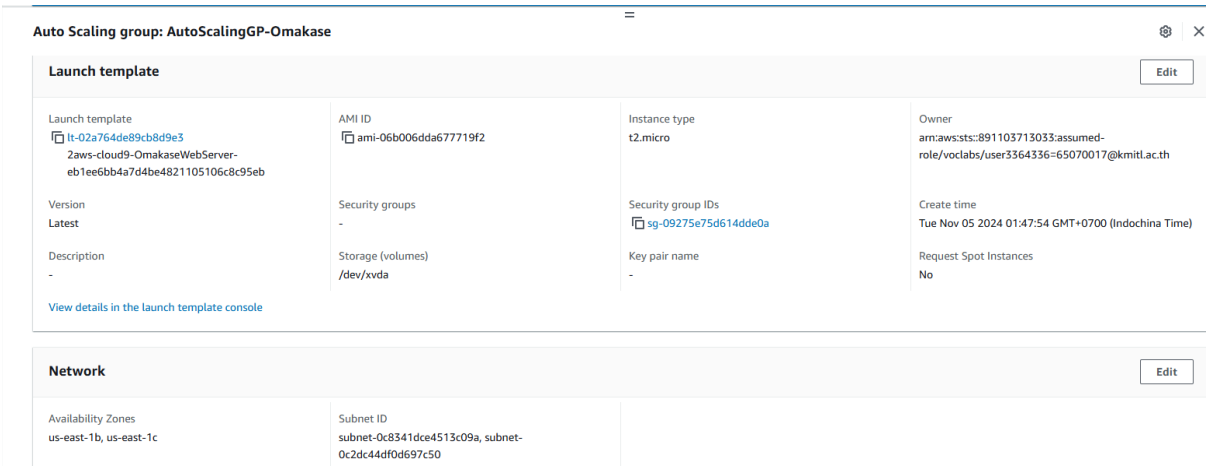
รูปที่ 4.14 Elastic Load Balancing

3. Auto Scaling

ช่วยตรวจสอบปริมาณการใช้งาน Server โดยเมื่อมีปริมาณการใช้งานที่มากเกินกว่าที่กำหนดจะมีการเพิ่ม EC2 Instance เข้ามาเพื่อลดภาระการทำงานของ EC2 Instance



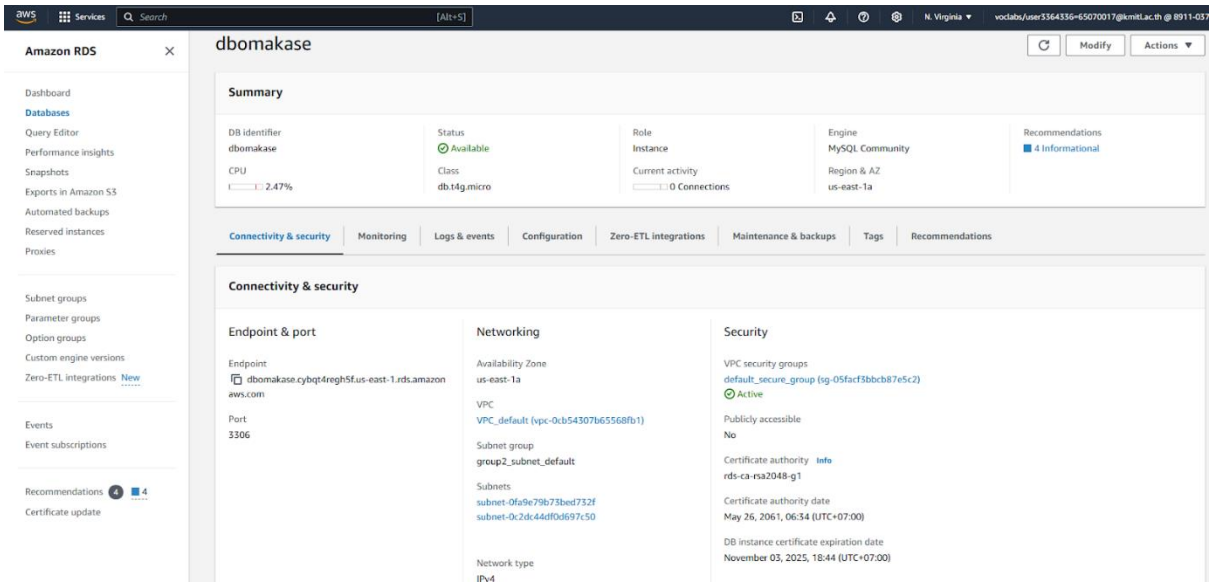
รูปที่ 4.15 Auto Scaling



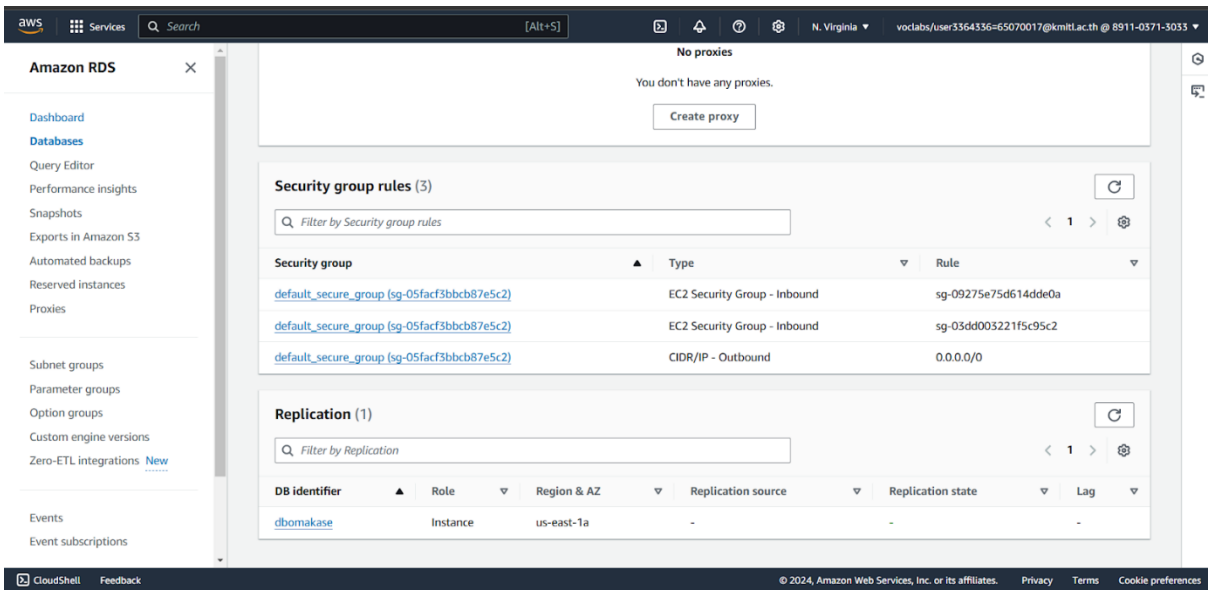
รูปที่ 4.16 Auto Scaling

4. Amazon Relational Database Service (Amazon RDS)

ใช้สร้าง SQL database ในการเก็บข้อมูลต่างๆ ของเว็บไซต์ตามใจเชฟ เช่น รายละเอียดแต่ละคอร์ส การจองคอร์สโอมากาเสะ



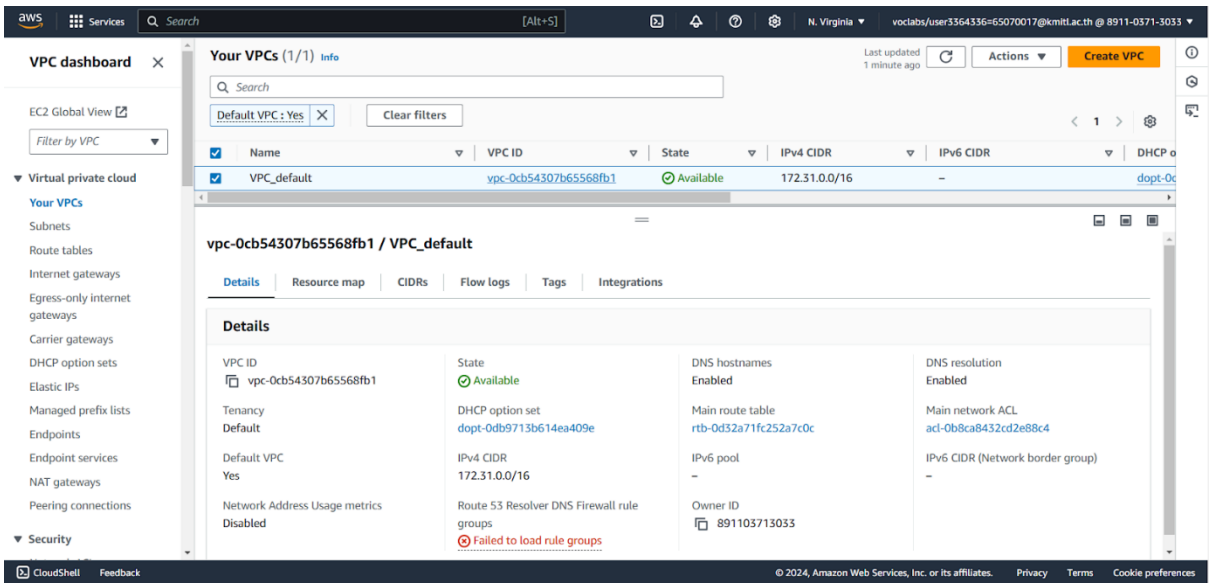
รูปที่ 4.17 Amazon Relational Database Service (Amazon RDS)



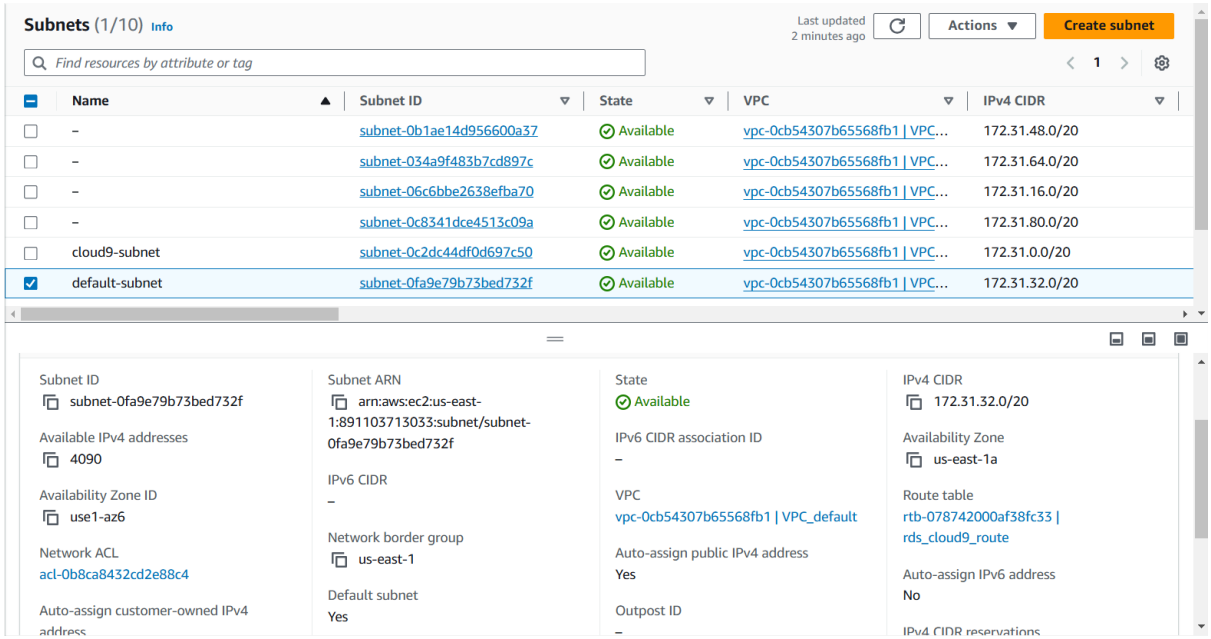
รูปที่ 4.18 Amazon Relational Database Service (Amazon RDS)

5. Amazon Virtual Private Cloud (Amazon VPC)

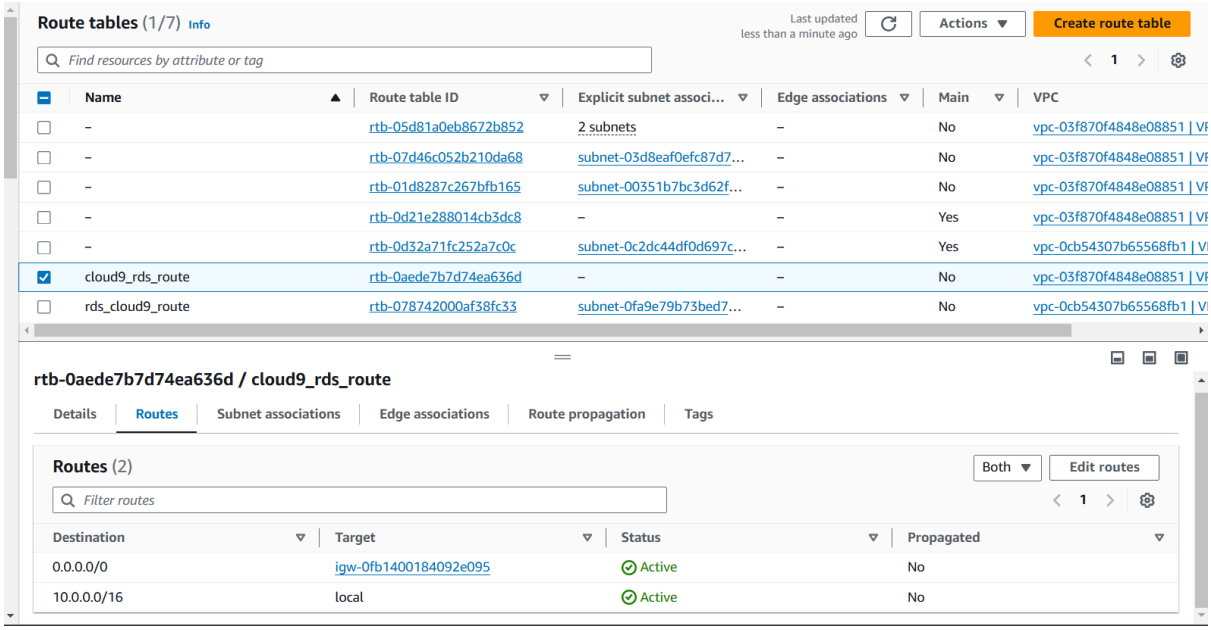
ใช้ VPC ในการควบคุมการเชื่อมต่อระหว่างเซิร์ฟเวอร์และฐานข้อมูลได้อย่างปลอดภัย รวมถึงการจัดการความปลอดภัยของข้อมูลที่สำคัญ เช่น ข้อมูลผู้ใช้ และข้อมูลการชำระเงิน ทำให้มั่นใจได้ว่าข้อมูลเหล่านี้จะได้รับการปกป้องจาก การเข้าถึงที่ไม่ได้รับอนุญาต



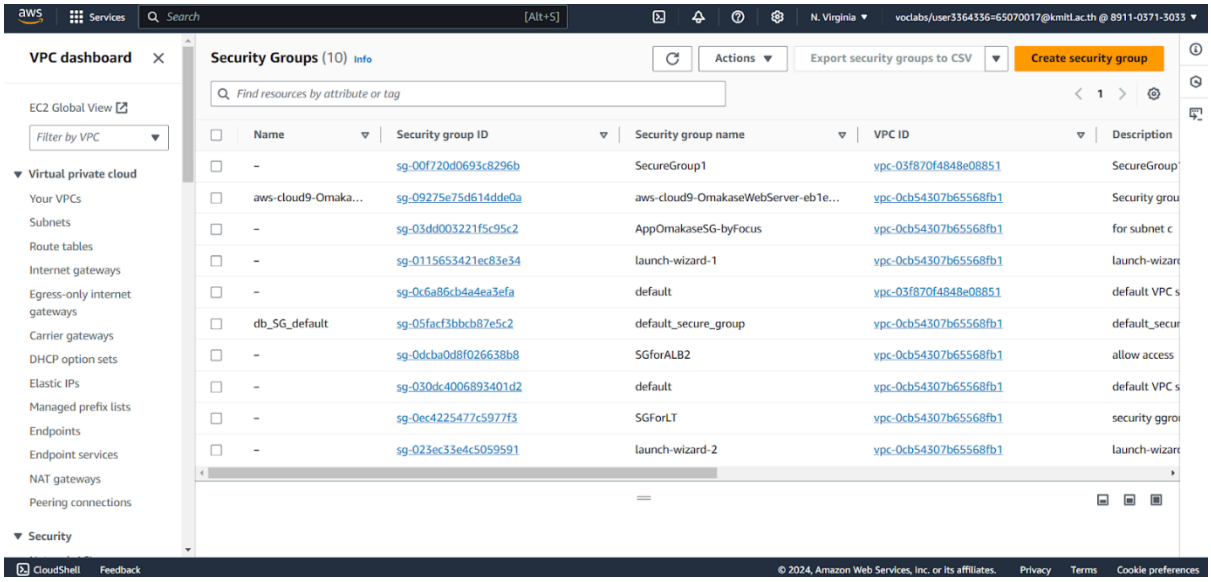
รูปที่ 4.19 Amazon Virtual Private Cloud (Amazon VPC)



รูปที่ 4.20 Amazon Virtual Private Cloud (Amazon VPC)



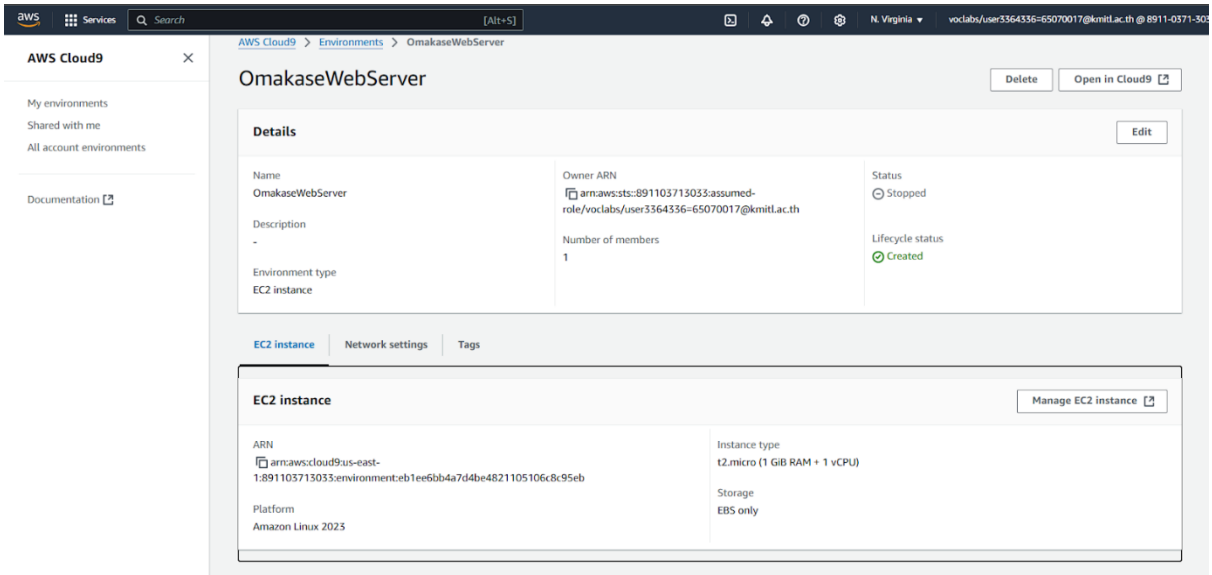
รูปที่ 4.21 Amazon Virtual Private Cloud (Route tables)



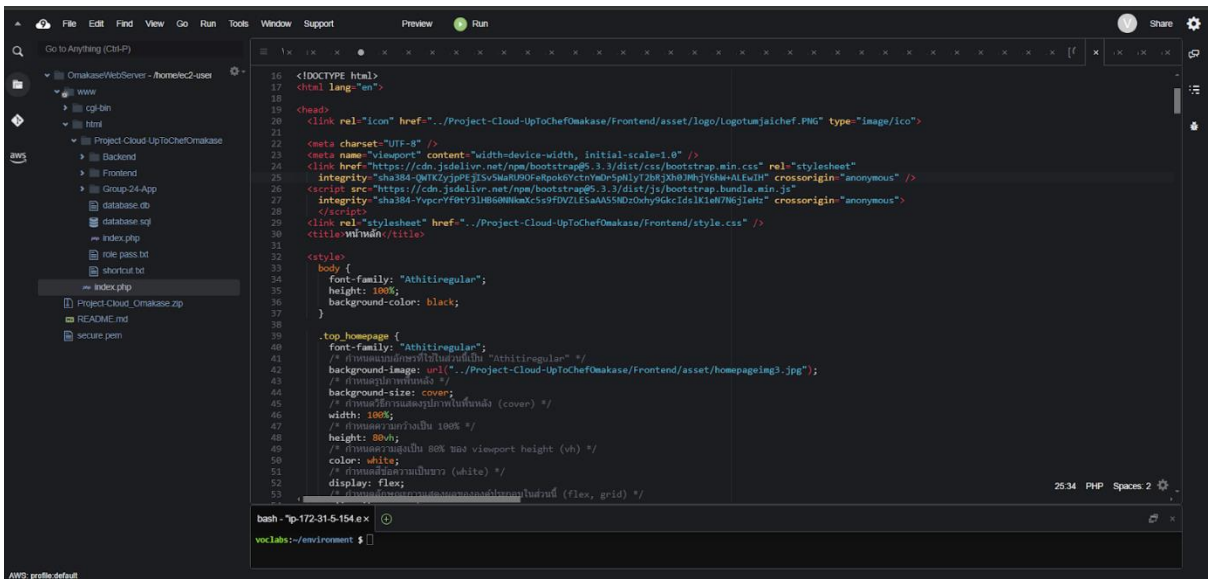
รูปที่ 4.21 Amazon Virtual Private Cloud (Security Groups)

6. Amazon Cloud9

สภาพแวดล้อมสำหรับการพัฒนาแบบเบ็ดเสร็จแบบ Cloud (IDE) ใช้งาน และแก้จุดบกพร่องโค้ดโดยใช้เบราว์เซอร์ ใช้ในการ Deploy website นำ Code ขึ้นไปบน Cloud และเชื่อมต่อกับ Database ขึ้น cloud โดยมีการ setting เปิดใช้งานตัว Apache และสามารถใช้ Cloud9 ในการแก้ไฟล์เว็บเซิฟได้ ทำให้เขียน ใช้งานและเข้าถึงข้อบกพร่องได้อย่างง่ายดาย ทั้งยังช่วยเพิ่มประสิทธิภาพในการเริ่มต้นโปรเจกต์อีกด้วย



รูปที่ 4.22 Amazon Cloud9



รูปที่ 4.23 Amazon Cloud9

บทที่ 5

สรุปผลการดำเนินงาน

5.1 ที่มาและการทำงานโดยรวมของแอปพลิเคชัน

โอมากาเสะเป็นร้านอาหารที่มีความละเอียด ใส่ใจในขั้นตอนการทำอาหารและมีพิธีรีตอง สูง รายการอาหารในคอร์สโอมากาเสะจะเปลี่ยนแปลงไปตามแต่ละโอกาสหรือฤดูกาลที่เหมาะสม โดยทางร้านจะเป็นผู้ตัดสินใจเพียงฝ่ายเดียวว่าจะทำรายการอาหารอะไรบ้างระหว่างคอร์สนั้น เราเล็งเห็นว่าร้านโอมากาเสะมีรูปแบบร้านที่น่าสนใจในด้านการบริการที่มีแนวคิดในการ เสิร์ฟแบบ “ตามใจเชฟ” ซึ่งตรงกับความหมายของคำว่า “โอมากาเสะ” ในภาษาญี่ปุ่นนั่นเอง ขยายความคือ การรับประทานอาหารโดยที่ลูกค้าไม่ได้เลือกรายการเอง แต่ละรายการที่ เสิร์ฟเชฟจะเป็นคนจัดให้ โดยเราได้นำวิธีการนำเสนออาหารแบบโอมากาเสะและพัฒนาเพิ่มเติมเป็นรูปแบบของแอปพลิเคชัน ร้าน ที่สามารถควบคุมระบบต่างๆภายในร้านทั้งการจัดการรายการอาหารทั้งหมดในร้าน, จัดการ คอร์สโอมากาเสะ, ดูรายการทั้งหมดที่ลูกค้าจอง, รับชำระเงินสุทธิ, จองคอร์สโอมากาเสะ, ชำระเงินมัดจำ, ยกเลิกการจอง และให้ลูกค้าสามารถตรวจสอบคอร์สโอมากาเสะที่จองไว้ได้

5.2 ประโยชน์ของแอปพลิเคชัน

ส่วนของลูกค้าสามารถจองโต๊ะ ดูอาหารแต่ละคอร์ส และชำระเงินผ่านแอปพลิเคชันได้อย่างง่ายดายโดยไม่ต้องรอการติดต่อจากร้าน

ส่วนของผู้จัดการร้านช่วยลดความซับซ้อนในการจัดการร้าน ทั้งในเรื่องการจัดการเพิ่ม ลบ เมนูอาหารในแต่ละคอร์ส การจอง การชำระเงิน และการสื่อสารกับลูกค้า

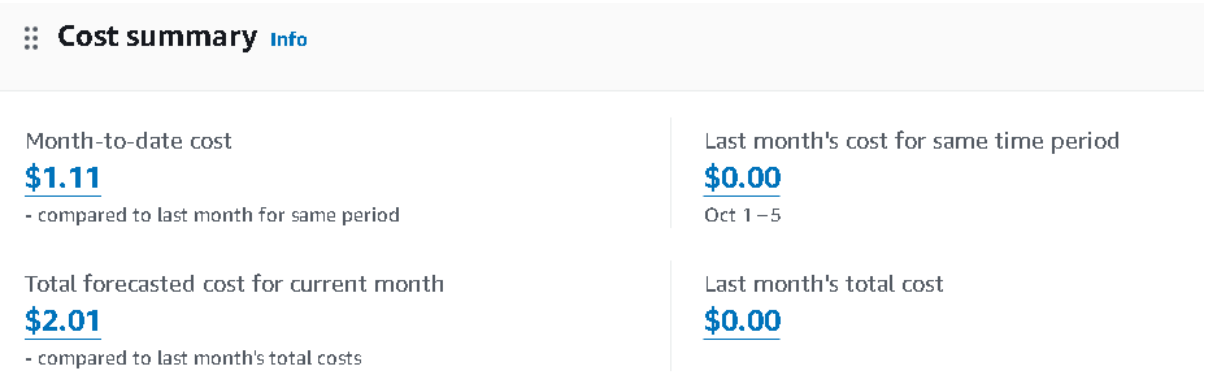
5.3 ปัญหาและอุปสรรคที่เกิดขึ้นในการพัฒนา

- เนื่องจากภาระงานที่มีมากตลอดเทอมการศึกษาทำให้ต้องแบ่งเวลาไปจัดการโครงงานอื่นด้วยๆ ส่งผลให้ไม่สามารถทุ่มเทเวลากับโครงงานนี้ได้เท่าที่ควร
- การขาดความรู้ความเข้าใจในบาง Service ทำให้ในส่วนของการปฏิบัติมีปัญหาติดขัด และต้องกลับมาแก้ไขให้การทำงานสอดคล้องกันส่งผลให้การทำงานล่าช้า

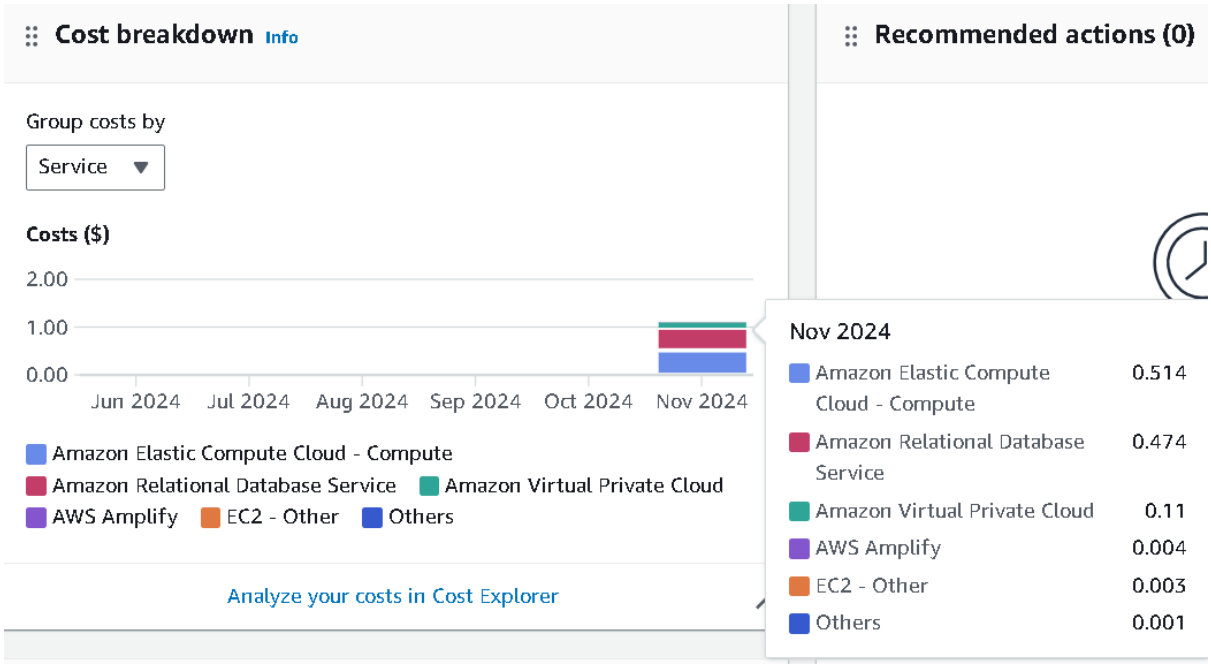
5.4 ข้อจำกัดระบบ

ระบบปัจจุบันออกแบบมาให้รองรับการทำงานพื้นฐาน หากในอนาคตหากมีการพัฒนา Feature ที่มากขึ้นจะต้องมีการจัดการ Service ต่างๆ เพิ่มเติมให้รองรับการทำงานต่างๆ ซึ่งจะส่งผลให้ค่าใช้จ่ายในการใช้งาน AWS เพิ่มขึ้น

5.5 ประมาณการค่าใช้จ่าย



รูปที่ 5.1 ประมาณการใช้จ่าย



รูปที่ 5.2 ประมาณการใช้จ่าย

5.6 การวิเคราะห์และออกแบบเพิ่มเติมโดยไม่คำนึงถึงค่าใช้จ่าย

เนื่องจากเว็บตามใจเชฟเป็นเว็บที่ถูกพัฒนาขึ้นโดยมีจุดประสงค์หลัก คือ การจองคอร์สโอมากาเสะ ซึ่งถ้าออกแบบโดยไม่คำนึงถึงปริมาณค่าใช้จ่ายจะมี Business Requirement เพิ่มเติมดังนี้

1. ตรวจสอบ อัพเดตสถานะ จัดเก็บหลักฐานการชำระเงิน
    - Amazon Payment Cryptography : เพิ่มความปลอดภัยในการชำระเงินผ่านการเข้ารหัสสำหรับข้อมูลสำคัญ เช่น PayPal และบัตรเครดิต
    - Amazon API Gateway : เพิ่ม ลบ แก้ไข เรียกดูสถานะการจอง
    - Amazon Lambda : ตรวจสอบสถานการชำระเงินจากผู้ให้บริการการเงิน เช่น ธนาคาร
    - Amazon S3 : เก็บรูปภาพ เช่น สลิปการชำระเงิน
2. เพิ่มความปลอดภัย
    - NAT Gateway