

สถานการณ์ที่ 4: การลดขยะอาหารจากของที่ใกล้หมดอายุในครัวเรือนและชุมชนอย่างมีระบบ

รายชื่อสมาชิก :

นางสาวสุพนิดา	ยมภา	6609490047
นายกมลพันธ์	กันธายอด	6609520116
นายกীরติกร	บุญเจริญ	6609611774
นายชนะพล	กังพิ้ว	6609611824
นายณรงค์กรณ์	มงคลสวะไชย	6609611881
นายณัฐศักดิ์	ชนมนัส	6609611931
นางสาวอภิษฎา	ไชยเสนา	6609612301
นายธนกร	พยัคฆรักษ์	6609681223

ชื่อระบบ : BiteBright

ที่มาและความสำคัญ :

ในปัจจุบัน เรามักประสบปัญหาลืมนำวัตถุดิบหรืออาหารที่ซื้อไว้จนใกล้หมดอายุหรือหมดอายุโดยไม่ทันตั้งตัว นำไปสู่การทิ้งอย่างเปล่าประโยชน์ และก่อให้เกิดปัญหา Food Waste ที่เพิ่มมากขึ้น แม้อาจดูเป็นเรื่องเล็กน้อย แต่หากสะสมในภาพรวมแล้ว จะส่งผลกระทบต่อสิ่งแวดล้อมอย่างมีนัยสำคัญ

จากปัญหาดังกล่าว ทีมผู้พัฒนาจึงได้ออกแบบระบบ “ BiteBright ” ระบบจัดการอาหารและวัตถุดิบในครัวเรือนแบบอัจฉริยะ ที่ช่วยให้ผู้ใช้สามารถใช้วัตถุดิบให้ทันก่อนหมดอายุ ด้วยการแจ้งเตือนล่วงหน้าผ่าน Line นอกจากนี้หากผู้ใช้ไม่ทราบว่าจะนำวัตถุดิบไปประกอบอาหารเป็นเมนูใด ระบบยังสามารถแนะนำเมนูอาหารที่สอดคล้องกับวัตถุดิบที่มีอยู่ และยังสามารถตรวจสอบรายการของกินผ่านระบบได้ทุกที่ทุกเวลา เพื่อลดการซื้อของซ้ำโดยไม่จำเป็น

ระบบไม่เพียงช่วยลดปริมาณขยะอาหาร แต่ยังส่งเสริมพฤติกรรมการบริโภคอย่างมีสติ สร้างจิตสำนึกด้านสิ่งแวดล้อม และสามารถต่อยอดเป็นเครื่องมือในการบริหารจัดการทรัพยากรในครัวเรือนอย่างมีประสิทธิภาพในระยะยาว

รายละเอียดโดยสังเขป :

BiteBright เป็น Web Application ที่ช่วยให้ผู้ใช้งานสามารถบริหารจัดการอาหารและวัตถุดิบภายในบ้านได้ง่ายขึ้น ลดการเกิดการเกิดอาหารหมดอายุโดยไม่รู้ตัว และลดการซื้อซ้ำโดยไม่จำเป็น โดยมีฟังก์ชันสำคัญในการแจ้งเตือนก่อนวันหมดอายุผ่าน Email และระบบแนะนำเมนูอาหารจากวัตถุดิบที่มีอยู่

ปัญหาที่จะแก้ :

1. ของหมดอายุโดยไม่รู้ตัว
2. ซื้อซ้ำโดยไม่จำเป็น
3. ไม่มีระบบแจ้งเตือนของหมดอายุ
4. ไม่รู้จะทำเมนูอะไรจากวัตถุดิบที่มี

แนวทางการแก้ไข :

1. ระบบการบันทึกและแจ้งเตือนวันหมดอายุ
 - ผู้ใช้ ป้อนข้อมูลวัตถุดิบ/อาหารและ วันหมดอายุ ระบบจะเก็บข้อมูลไว้บนฐานข้อมูล และใช้ฟังก์ชัน Lambda แจ้งเตือนผ่าน Email ล่วงหน้าก่อนวันหมดอายุ
2. ลดการซื้อซ้ำโดยไม่จำเป็น
 - ระบบจะแสดงรายการวัตถุดิบที่มีอยู่ในบ้านแบบชัดเจน เพื่อให้ผู้ใช้เห็นว่าไม่จำเป็นต้องซื้อซ้ำ
3. แนะนำเมนูจากวัตถุดิบที่มี
 - ระบบจะสำรวจวัตถุดิบในครัวเรือนของผู้ใช้ และแนะนำเมนูที่สามารถทำได้โดยอิงจากฐานข้อมูลเมนู

แผนการพัฒนาระบบ :

- **รูปแบบของระบบ**

กลุ่มของเราจะพัฒนา web application สำหรับจัดการข้อมูลวัตถุดิบและอาหาร โดยผู้ใช้งานสามารถบันทึกข้อมูลวัตถุดิบ/อาหารได้ ระบุวันหมดอายุ ดูข้อมูลวัตถุดิบ และได้รับการแจ้งเตือนเมื่อวัตถุดิบใกล้หมดอายุผ่าน email
- **Frontend**
 - HTML, CSS, Javascript -> ใช้สำหรับพัฒนา interface ของ web application
 - Figma -> ใช้สำหรับการออกแบบ UX/UI

- **Backend**

สร้าง API โดยใช้ AWS API Gateway เพื่อเชื่อมต่อกับ Lambda functions ที่จะทำหน้าที่จัดการกับข้อมูลผู้ใช้ เช่น การเพิ่ม ลบ และเรียกดูวัตถุดิบที่มีในบ้าน รวมถึงข้อมูลเมนูอาหารจากฐานข้อมูล

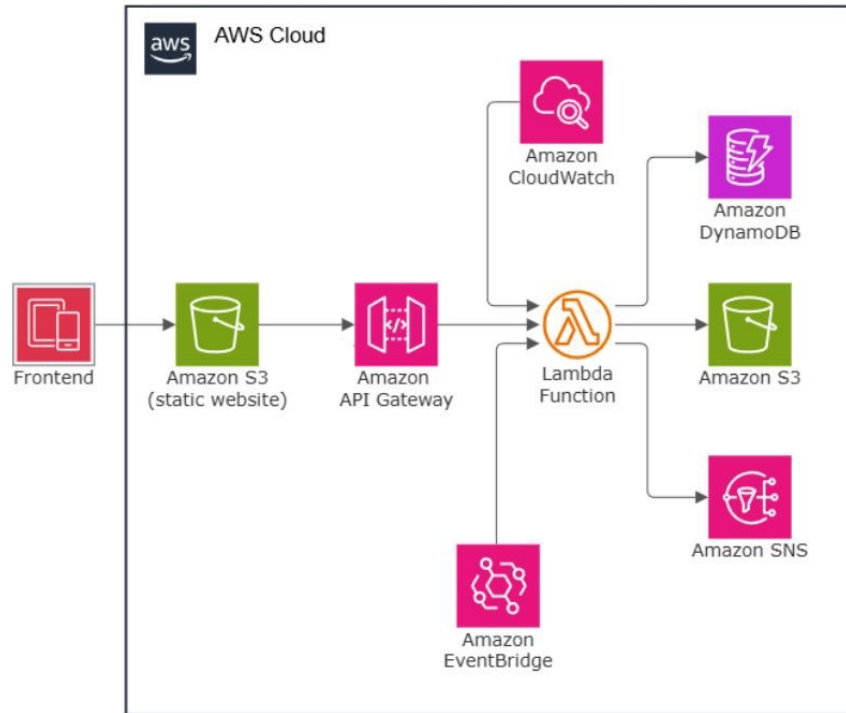
- **ระบบแจ้งเตือน**

เราจะสร้าง Lambda Function สำหรับตรวจสอบวันหมดอายุของวัตถุดิบ และใช้ AWS SNS เพื่อส่งคำสั่งไปยัง Email ให้เตือนเมื่อใกล้ถึงวันหมดอายุ

Service ที่เลือก :

- AWS Lambda สำหรับประมวลผลคำสั่งต่าง ๆ เช่น สร้างข้อความแจ้งเตือน แนะนำเมนูอาหาร
- AWS SNS ใช้เพื่อส่ง event แจ้งเตือนกระจายไปยัง Lambda หรือระบบที่เชื่อมต่อ
- AWS DynamoDB ใช้ในการเก็บข้อมูล latency ต่ำ เข้าถึงข้อมูลได้ไวกแบบ real time เก็บข้อมูล 2 ส่วน คือ ข้อมูลวัตถุดิบที่ผู้ใช้มี กับ ข้อมูลเมนูอาหาร
- AWS API gateway ใช้เชื่อมต่อ frontend และ backend
- Amazon EventBridge ตั้งเวลาสำหรับแจ้งเตือนของใกล้หมดอายุ

บริการ AWS ที่ใช้ :



ระบบนี้ออกแบบมาเพื่อรองรับการรับข้อมูลจากผู้ใช้ผ่าน Web Application ที่โฮสต์บน Amazon S3 (Static Website Hosting) ซึ่งให้บริการหน้าเว็บแบบ static สำหรับการกรอกข้อมูล และโต้ตอบกับผู้ใช้

เมื่อผู้ใช้ส่งคำขอ (Request) ผ่านหน้าเว็บ คำขอทั้งหมดจะถูกส่งผ่านโปรโตคอล HTTPS ไปยัง Amazon API Gateway ซึ่งเป็นตัวกลางรับคำขอและจัดการ endpoint รวมถึงการควบคุมการเข้าถึง API ในส่วนของการตรวจสอบสิทธิ์ (Authentication & Authorization) ระบบจะใช้ Token ที่ฝัง userId โดย Token นี้จะถูกแนบมากับคำขอ และ Lambda Function จะทำหน้าที่ตรวจสอบความถูกต้องของ Token พร้อมดึง userId ไปใช้ในการประมวลผลข้อมูล หลังจากการยืนยันสิทธิ์สำเร็จ API Gateway จะเรียกใช้ AWS Lambda Function ซึ่งทำหน้าที่เป็นแกนหลักของระบบ (Backend Logic) โดย Lambda จะทำการประมวลผลข้อมูล เช่น การตรวจสอบความถูกต้องของข้อมูล การเขียนข้อมูลลงใน Amazon DynamoDB (ฐานข้อมูล NoSQL) และการจัดการไฟล์แนบ เช่น รูปภาพหรือเอกสาร โดยบันทึกไฟล์เหล่านั้นลงใน Amazon S3

สำหรับการแจ้งเตือนผู้ใช้เกี่ยวกับคำร้องหรือข้อมูลใหม่ ระบบจะใช้ Amazon SNS (Simple Notification Service) ในการส่งการแจ้งเตือนทางอีเมลไปยังผู้ที่เกี่ยวข้อง ซึ่ง SNS ทำหน้าที่เป็นตัวกลางรับข้อความจาก Lambda และกระจายไปยัง Email Subscriber

นอกจากนี้ ระบบยังสามารถใช้ Amazon EventBridge เพื่อตั้งเวลา (Scheduled Task) ให้ AWS Lambda ทำงานอัตโนมัติตามเวลาที่กำหนด เช่น ตรวจสอบข้อมูลใกล้เคียงดอทคอมเป็นประจำทุกวัน โดยไม่ต้องมีการสั่งการจากผู้ใช้โดยตรง และระบบจะใช้ Amazon CloudWatch ในการเก็บ Log การทำงานของ Lambda, API Gateway และบริการอื่น ๆ เพื่อให้ผู้ดูแลระบบสามารถตรวจสอบการทำงาน วิเคราะห์ปัญหา และตั้งค่าการแจ้งเตือน (Alert) ในกรณีที่เกิดเหตุการณ์ผิดปกติ เช่น Lambda ล้มเหลวหรือ API มี Error Responses เกินค่าที่กำหนด

Use cases :

กรณีที่ 1: ผู้ใช้ลงทะเบียนและเข้าสู่ระบบสำเร็จ

- สถานการณ์: ผู้ใช้ใหม่ลงทะเบียนบัญชีและเข้าสู่ระบบ
- ขั้นตอนการทดสอบ:
 - ไปยังหน้า register.html
 - กรอก username, email, password และ confirm password
 - กด "Register"
 - หลังสำเร็จ กดไปที่ login.html
 - กรอก username, password และกด "Login"
- ผลลัพธ์ที่คาดหวัง:
 - แสดง Toast "Register success!"
 - Redirect ไปหน้า Login
 - เข้าสู่ระบบสำเร็จ และแสดงชื่อผู้ใช้และอีเมลใน dropdown

- ผ่านการทดสอบ:

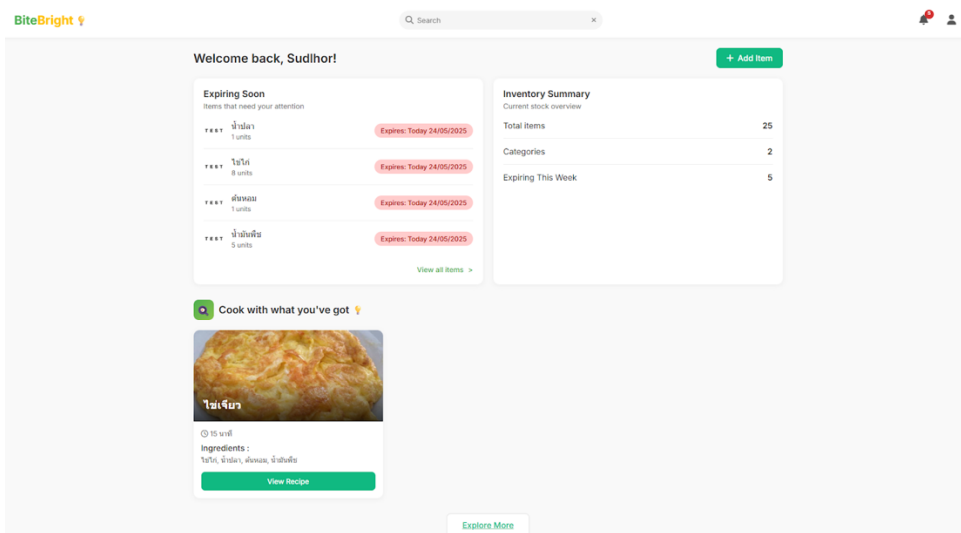
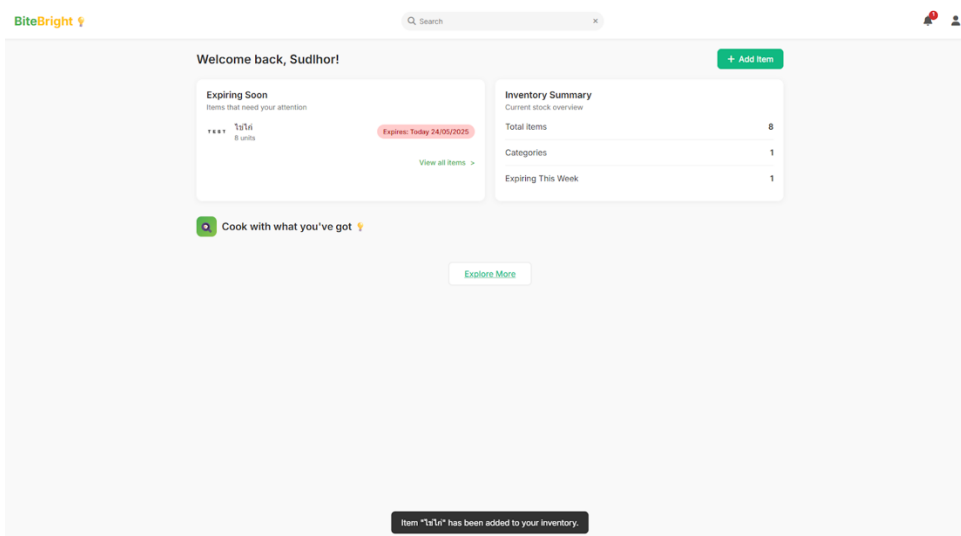
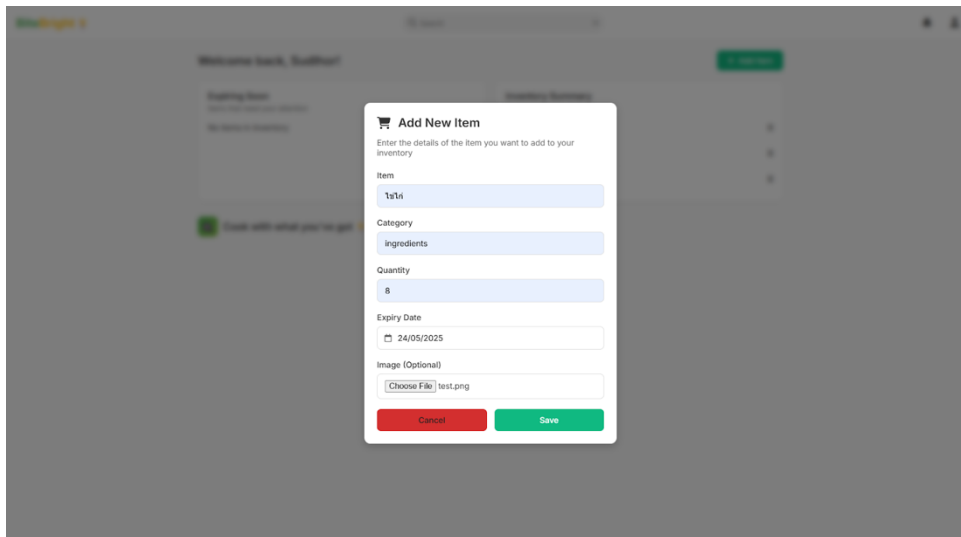
A screenshot of a web browser showing the registration page for BiteBright. The browser's address bar displays "webapporigintest.s3-website-us-east-1.amazonaws.com/register.html". The page features the BiteBright logo and the tagline "Manage your food inventory smartly". Below this, the heading "Create your account" is followed by a form with fields for "first", "email" (containing "alaha.nga@some.ku.ac.th"), and "password" (with masked characters). A green "Sign Up" button is at the bottom of the form, with a link "Already have an account? Login" below it.

A screenshot of a web browser showing the login page for BiteBright. The browser's address bar displays "webapporigintest.s3-website-us-east-1.amazonaws.com/login.html". The page features the BiteBright logo and the tagline "Manage your food inventory smartly". Below this, the heading "Login to your account" is followed by a form with fields for "first" and "password" (with masked characters). A green "Login" button is at the bottom of the form, with a link "Don't have an account? Sign up" below it. At the bottom of the page, a black notification box with white text reads "Login success! Redirecting...".

กรณีที่ 2: ผู้ใช้เพิ่มรายการอาหารและได้รับการแนะนำเมนู

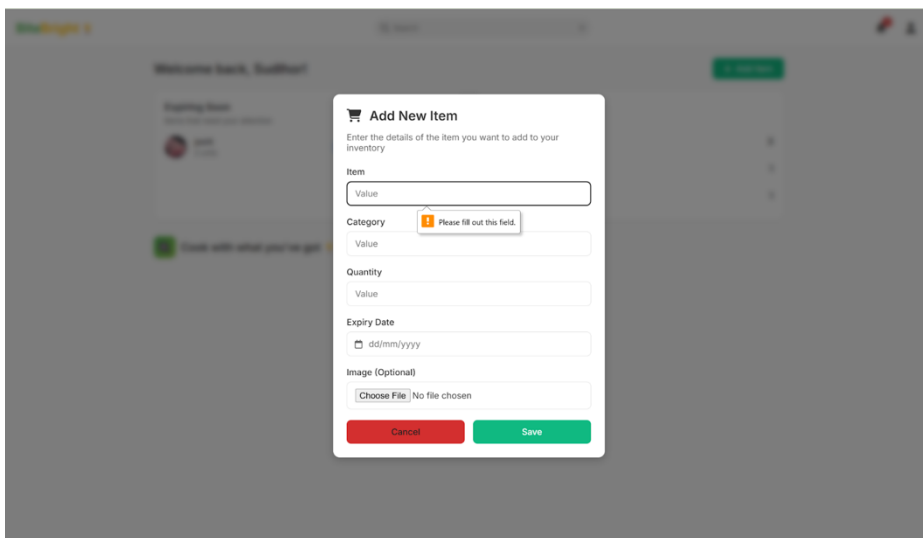
- สถานการณ์: ผู้ใช้เข้าสู่ระบบแล้วเพิ่มวัตถุดิบในครัวและได้รับคำแนะนำเมนู
- ขั้นตอนการทดสอบ:
 - เข้าสู่ระบบ
 - ไปที่หน้า "เพิ่มวัตถุดิบ" (Add Item)
 - กรอกชื่อวัตถุดิบ: ไข่ไก่, วันหมดอายุ, ปริมาณ ฯลฯ แล้วกด "Add"
 - กรอกชื่อวัตถุดิบ: น้ำปลา, วันหมดอายุ, ปริมาณ ฯลฯ แล้วกด "Add"
 - กรอกชื่อวัตถุดิบ: ต้นหอม, วันหมดอายุ, ปริมาณ ฯลฯ แล้วกด "Add"
 - กรอกชื่อวัตถุดิบ: น้ำมันพืช, วันหมดอายุ, ปริมาณ ฯลฯ แล้วกด "Add"
 - ไปที่หน้า "เมนูแนะนำ" (Recommend Menu)
- ผลลัพธ์ที่คาดหวัง:
 - Toast: "Item added successfully"
 - แสดงเมนูที่สามารถทำได้โดยใช้วัตถุดิบที่ "Add" เข้าไป เช่น "ไข่เจียว"

- ผ่านการทดสอบ:



กรณีที่ 3: ผู้ใช้กรอกข้อมูลไม่ครบในฟอร์มเพิ่มวัตถุดิบ (กรณีข้อผิดพลาด)

- สถานการณ์: ผู้ใช้ลืมกรอกชื่อวัตถุดิบแล้วกดปุ่มเพิ่ม
- ขั้นตอนการทดสอบ:
 - เข้าสู่ระบบ ไปที่หน้า "เพิ่มวัตถุดิบ"
 - ไม่กรอกชื่อวัตถุดิบ แล้วกด "Add"
- ผลลัพธ์ที่คาดหวัง:
 - ไม่เพิ่มรายการเข้าไปในระบบ
 - แสดงข้อความผิดพลาด เช่น "Please enter item name"
 - ไม่มีการ redirect หรือ crash
- ผ่านการทดสอบ (ระบบรับมือข้อผิดพลาดได้อย่างถูกต้อง)



The screenshot shows a web application interface with a modal form titled "Add New Item". The form contains the following fields and elements:

- Item:** A text input field with a placeholder "Value".
- Category:** A dropdown menu with a placeholder "Value". A red error message "Please fill out this field." is displayed next to it.
- Quantity:** A text input field with a placeholder "Value".
- Expiry Date:** A date input field with a placeholder "dd/mm/yyyy".
- Image (Optional):** A file upload area with a "Choose File" button and the text "No file chosen".
- Buttons:** "Cancel" (red) and "Save" (green) buttons at the bottom.

ข้อดีและข้อเสียของระบบ :

- ข้อดี

- ใช้ AWS Lambda ซึ่งดูแลเรื่อง infrastructure โดยอัตโนมัติ
- สามารถ deploy อัปเดตได้อย่างรวดเร็ว
- AWS ให้ความเสถียรสูง ทั้ง Lambda และ DynamoDB
- ใช้ DynamoDB ซึ่งมี latency ต่ำ
- Lambda สามารถสเกลตามคำร้องขอ
- ลดผลกระทบต่อสิ่งแวดล้อม เนื่องจากลดขยะจากอาหารหรือของหมดอายุ
- ใช้ทรัพยากรเท่าที่จำเป็น

- ข้อเสีย

- ต้องมีการจัดการ version ของ Lambda อย่างเป็นระบบ
- โค้ดที่ SNS หมด จะส่งผลการแจ้งเตือน
- การแนะนำเมนูต้อง optimize เพื่อไม่ใช้เวลาในการประมวลผลมาก
- ค่าใช้จ่ายอาจเพิ่มขึ้นหากมีผู้ใช้จำนวนมากและมีการเรียก Lambda บ่อย
- ควรออกแบบ UI/UX ให้ส่งเสริมการใช้งานระยะยาว เพื่อความยั่งยืนในการใช้งาน

URL ของระบบ :

- Web App BiteBright :

<http://webappbrigtbrittenow.s3-website-us-east-1.amazonaws.com>

- Video Demo :

<https://1drv.ms/f/c/811ccc5588f65c4f/EpJJE4hlvkREi7Ext2fPGo0BsZWhgeLT3dD1OaTI3q05PO?e=8pkdix>

สรุปผลการดำเนินโครงการ (bitebright)

โครงการนี้มีเป้าหมายเพื่อพัฒนาเว็บแอปพลิเคชันที่ช่วยให้ผู้ใช้งานสามารถจัดการวัตถุดิบอาหารในครัวได้อย่างมีประสิทธิภาพ โดยโฟกัสที่การลดการสูญเสียอาหารจากการหมดอายุ โดยฟีเจอร์หลักที่พัฒนาเสร็จสมบูรณ์ได้แก่ :

1. ระบบลงทะเบียนและเข้าสู่ระบบ

- รองรับการลงทะเบียนด้วย `username`, `email` และ `password`
- เข้าสู่ระบบและบันทึก session ด้วย `userId` ผ่าน LocalStorage

2. ระบบเพิ่มรายการอาหาร (Add Item)

- ผู้ใช้สามารถเพิ่มวัตถุดิบ เช่น ชื่อวัตถุดิบ, ปริมาณ, วันหมดอายุ
- เชื่อมต่อกับฐานข้อมูล DynamoDB ผ่าน AWS Lambda

3. ระบบแนะนำเมนูอาหาร (Recommend Menu)

- แนะนำเมนูอาหารจากวัตถุดิบที่มีอยู่ โดยพิจารณาจากวัตถุดิบที่ใกล้หมดอายุก่อน

4. ระบบการแจ้งเตือน (Notification)

- แจ้งเตือนเมื่อวัตถุดิบใกล้หมดอายุ
- มี UI dropdown ที่แสดงสถานะแจ้งเตือน และสามารถกด "Mark all as read" ได้

5. UI/UX ที่ใช้งานง่าย

- ใช้การออกแบบแนวสว่างและเรียบง่าย (Bright & Minimal)
- ใช้งานได้ทั้งบน Desktop และ Mobile

6. Deployment

- ระบบถูก Deploy ผ่าน AWS (Lambda, API Gateway, DynamoDB)

แนวทางพัฒนาต่อในอนาคต

1. เพิ่มระบบยืนยันตัวตน (Authentication) ที่ปลอดภัยขึ้น
 - ใช้ JWT หรือ OAuth2 แทน LocalStorage
 - รองรับการเข้าสู่ระบบผ่านบัญชี LINE หรือ Google
2. เพิ่มฟีเจอร์ Scan Barcode หรือถ่ายรูปวัตถุดิบ
 - เพื่อให้ผู้ใช้เพิ่มวัตถุดิบได้รวดเร็วยิ่งขึ้น
3. เพิ่มระบบวิเคราะห์พฤติกรรมการใช้อาหาร
 - วิเคราะห์วัตถุดิบที่ใช้งานบ่อย
 - สร้างกราฟแสดงปริมาณอาหารที่สูญเสียเปล่ารายเดือน
4. รองรับการใช้งานแบบหลายผู้ใช้ร่วมกัน (Multi-user/Family Mode)
 - สมาชิกในครัวสามารถใช้งานร่วมกันได้
5. แปลงระบบเป็น Progressive Web App (PWA)
 - เพื่อให้สามารถใช้งานแบบ offline และติดตั้งบนมือถือได้
6. เชื่อมต่อกับ AI แนะนำเมนูชั้นสูง
 - แนะนำเมนูโดยวิเคราะห์จากวัตถุดิบ, ความชอบของผู้ใช้ และข้อมูลโภชนาการ