

สถานการณ์ที่ 1

การจัดการข้อมูลผู้สูญหายผ่านเพจโซเชียลมีเดียอย่างมีระบบ

ระบบ Lost & Found

รายชื่อสมาชิก

ชญานุช อิ่มเกิด	6609650277
ณัชร ดานุสวัสดิ์	6609650319
ดวงกมล แก้วศาลาภูมิ	6609650384
ปิณฑร จาดเนื่อง	6609650483
พีรดา งามวงศ์วิโรจน์	6609650533
ภกีน มณีรัตน์	6609650558
สิฏฐิยา สง่าเมือง	6609650681
อนุตตมา แสงทอง	6609650731

กลุ่มที่ 2

วิชา INTRODUCTION TO CLOUD COMPUTING TECHNOLOGY

CS332/232

ระบบ Lost & Found

ระบบ Lost & Found ถูกออกแบบมาเพื่อช่วยจัดการข้อมูลผู้สูญหายโดยมีการรวบรวมข้อมูล ลดความสับสนและซ้ำซ้อน อีกทั้งยังเพิ่มประสิทธิภาพในการค้นหา ติดตาม และมีเว็บไซต์ที่ทำให้การแจ้งเบาะแสสามารถใช้งานได้ง่ายยิ่งขึ้น

ที่มาและความสำคัญ

ในปัจจุบันปัญหาเกี่ยวกับการแจ้งข้อมูลผู้สูญหายผ่านทางโซเชียลมีเดียมักขาดการจัดการอย่างเป็นระบบ ขาดการจัดเก็บข้อมูลและขาดเครื่องมือที่ช่วยอำนวยความสะดวกในการค้นหาและติดตามสถานะ ส่งผลให้เกิดความล่าช้าในการค้นหาข้อมูล เกิดปัญหาข้อมูลซ้ำซ้อน ข้อมูลตกหล่น และความยากลำบากในการตรวจสอบความคืบหน้าของแต่ละเคส

ระบบ Lost & Found จึงถูกพัฒนาขึ้นเพื่อแก้ไขปัญหาที่กล่าวมาข้างต้น โดยอาศัยเทคโนโลยี Cloud และบริการจาก AWS เพื่อสร้างเว็บสำหรับรวบรวมการแจ้งข้อมูลผู้สูญหาย มีฟังก์ชันที่ช่วยให้ผู้ที่แจ้งเบาะแสสามารถติดตามความคืบหน้าของการตามหาผู้สูญหายได้อย่างสะดวกสบาย ลดความวุ่นวาย และประหยัดเวลามากขึ้น รวมถึงมีการใช้ Machine Learning ในการจัดลำดับความสำคัญของผู้สูญหายในแต่ละกรณี เพื่อเพิ่มโอกาสในการติดตามและช่วยเหลือผู้สูญหายอย่างมีประสิทธิภาพ

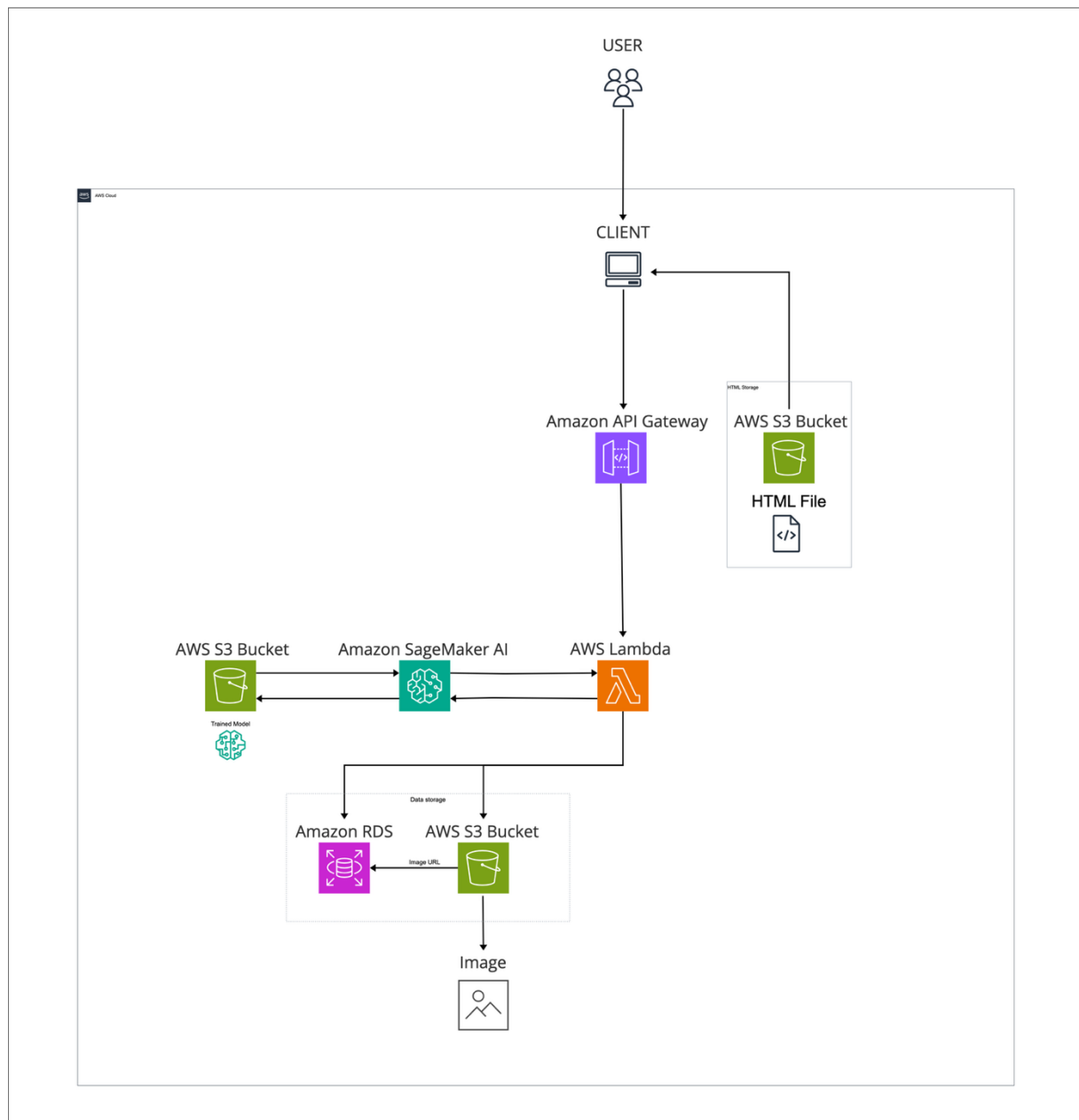
โครงการนี้จึงไม่เพียงช่วยให้การจัดการข้อมูลผู้สูญหายเป็นระบบมากขึ้น แต่ยังมีส่วนสำคัญในการส่งเสริมให้การแจ้งคนสูญหายถูกดำเนินการอย่างมีขั้นตอน สร้างความเชื่อมั่นในการติดตามความคืบหน้า และลดการสูญเสียที่อาจจะเกิดจากความล่าช้าหรือข้อมูลที่ตกหล่น อีกทั้งยังเป็นการประยุกต์ใช้เทคโนโลยีร่วมกับกระบวนการทำงานของมนุษย์ เพื่อช่วยลดความผิดพลาดที่อาจเกิดขึ้น และยกระดับประสิทธิภาพในการดำเนินงานในภาพรวม

กรณีการใช้งาน (Use cases) และประโยชน์สำหรับผู้ใช้งาน

Use Case ID	1
Use Case Name	ผู้ดูแล (Admin) แก้ไขข้อมูลผู้สูญหาย
Actor	ผู้ดูแลระบบ (Admin)
Purpose	เพื่อให้ Admin สามารถแก้ไขข้อมูลของผู้สูญหาย เพื่อความครบถ้วนและถูกต้องของข้อมูล
Pre conditions	<ul style="list-style-type: none"> - ต้องมีข้อมูลการเป็นผู้ดูแลระบบในฐานข้อมูล - ต้องมีข้อมูลผู้สูญหายที่ต้องการจะแก้ไขอยู่ในฐานข้อมูล
Post conditions	แก้ไขข้อมูลผู้สูญหายสำเร็จ
Main Flows	<ol style="list-style-type: none"> 1. เลือกข้อมูลของผู้สูญหายที่ต้องการแก้ไข 2. กดปุ่ม 'แก้ไข' 3. แก้ไขข้อมูลในส่วนที่ต้องการ (เช่น ชื่อ, อายุ, สถานะ, ระดับความเสี่ยง เป็นต้น) 4. กดปุ่ม 'ยืนยันการแก้ไข' 5. ระบบขึ้นแจ้งเตือนว่าเปลี่ยนแปลงสำเร็จ
Use Case ID	2
Use Case Name	ผู้ใช้แจ้งข้อมูลผู้สูญหาย
Actor	ผู้ดูแลระบบ (Admin), ผู้ใช้ (User)
Purpose	เพื่อให้ผู้ใช้สามารถแจ้งรายละเอียดผู้สูญหายในเว็บไซต์ได้
Pre conditions	<ul style="list-style-type: none"> - มีข้อมูลรายละเอียดของผู้สูญหาย - สามารถเข้าหน้าเว็บไซต์ได้
Post conditions	แจ้งข้อมูลผู้สูญหายสำเร็จ
Main Flows	<ol style="list-style-type: none"> 1. กดเลือกเมนู 'แจ้งผู้สูญหาย' 2. กรอกข้อมูลส่วนตัวของผู้ที่ต้องการแจ้งหาย (เช่น ชื่อ, อายุ, วันเกิด, ตาหนิ เป็นต้น) 3. กดปุ่ม 'ยืนยัน' 4. ระบบขึ้นแจ้งเตือนแสดงให้ทราบว่าบันทึกข้อมูลลงในฐานระบบสำเร็จ

Use Case ID	3
Use Case Name	ผู้ใช้งานค้นหาข้อมูลผู้สูญหาย
Actor	ผู้ดูแลระบบ (Admin), ผู้ใช้ (User)
Purpose	เพื่อให้ผู้ใช้งานสามารถค้นหารายละเอียดผู้สูญหายในเว็บไซต์ได้
Preconditions	<ul style="list-style-type: none"> - มีข้อมูลรายละเอียดของผู้สูญหาย - สามารถเข้าหน้าเว็บไซต์ได้
Post conditions	ค้นหาข้อมูลผู้สูญหายสำเร็จ
Main Flows	<ol style="list-style-type: none"> 1. กดเลือกเมนู 'ค้นหาผู้สูญหาย' 2. กรอกข้อมูลของผู้ที่ต้องการค้นหา (สามารถเป็นชื่อ, นามสกุล, อายุ, ความเสี่ยง) 3. ระบบแสดงผลลัพธ์ที่ตรงกับคำค้นหา 4. เลือกบุคคลที่ต้องการจากรายการ

สถาปัตยกรรมของระบบ (AWS Architecture Diagram)



ขั้นตอนการทำงานของระบบ AWS

1. เมื่อผู้ใช้เข้าสู่ เว็บไซต์ Lost and Found ไฟล์ HTML จะถูก โหลดและแสดงผลที่ฝั่ง Client โดยตรง จาก Amazon S3 Bucket
2. หากผู้ใช้ต้องการแจ้งข้อมูลบุคคลสูญหาย ข้อมูลจะถูกกรอกและ ส่งจากฝั่ง Client ไปยัง Amazon API Gateway ซึ่งทำหน้าที่เป็นจุดเชื่อมต่อและกำหนดเส้นทางข้อมูลไปยัง AWS Lambda

3. เมื่อ AWS Lambda ได้รับข้อมูล ระบบจะ ส่งข้อมูลดังกล่าวไปยัง Amazon SageMaker AI เพื่อประมวลผลและประเมินระดับความเสี่ยงของกรณีที่แจ้งเข้ามา โดยใช้ โมเดล AI ที่จัดเก็บอยู่ใน Amazon S3 Bucket โมเดล AI นี้จะทำการจำแนกระดับความเสี่ยงเป็น "สูง", "ปานกลาง" หรือ "ต่ำ"
4. ผลลัพธ์จาก SageMaker จะถูกส่งกลับมายัง Amazon Lambda ก่อนที่จะถูก บันทึกเข้าสู่ฐานข้อมูล Amazon RDS ส่วน ไฟล์ภาพของบุคคลสูญหายจะถูกอัปโหลดและจัดเก็บใน Amazon S3 Bucket และ URL ของภาพจะถูกบันทึกเชื่อมโยงในฐานข้อมูล RDS ด้วย
5. ในกรณีที่มีการค้นหาคคนสูญหาย ระบบจะ ดึงข้อมูลที่เกี่ยวข้องจากฐานข้อมูล Amazon RDS และนำมา แสดงผลบนหน้าเว็บไซต์ ให้กับผู้ใช้งาน

ขั้นตอนหลักที่สำคัญในการสร้างและตั้งค่า AWS Service

ขั้นตอนการสร้าง Amazon S3

1. เข้าสู่หน้า AWS Management Console แล้วค้นหาบริการ Amazon S3
2. เลือกแท็บ General purpose buckets และคลิก Create bucket ที่มุมขวาบน
3. ตั้งค่า Bucket type เป็น General purpose
4. ตั้งชื่อ Bucket เป็น lostfound123
5. ในหัวข้อ Block Public Access settings for this bucket ให้ยกเลิกการเลือกทั้งหมด และเลือก I acknowledge that the current settings might result in this bucket and the objects within becoming public.
6. คงค่าการตั้งค่าอื่นไว้ตามค่าเริ่มต้น (default) แล้วคลิก Create bucket

ขั้นตอนการสร้าง AWS Lambda

1. ในหน้า Console ค้นหาบริการ AWS Lambda
2. ไปที่แท็บ Functions และคลิก Create function
3. ตั้งชื่อฟังก์ชันว่า lostandfound และเลือก Runtime เป็น Node.js 22.x
4. คลิกที่ Change default execution role
5. ในหัวข้อ Execution role ให้เลือก Use an existing role
6. จากนั้นเลือก Existing role เป็น LabRole
7. คลิก Create function เพื่อสร้าง Lambda

ขั้นตอนการสร้าง Amazon API Gateway

1. ค้นหาบริการ API Gateway ผ่านหน้า Console
2. คลิก Create an API
3. เลือก HTTP API แล้วคลิก Build
4. ตั้งชื่อ API ในช่อง API name
5. ในหัวข้อ Integrations คลิก Add Integration และเลือก Lambda
6. กำหนด Region เป็น us-east-1
7. ในช่อง Lambda function ให้เลือกฟังก์ชัน Lambda ที่ได้สร้างไว้แล้วคลิก Next

8. กำหนด Route
9. เลือก Method: POST/GET
10. กำหนด Resource path: /missing_all
11. คลิก Next แล้วเลือก Deploy stage
12. กด Create เพื่อสร้าง API Gateway พร้อมใช้งาน
13. หลังจากสร้างเสร็จจะได้ ลิงค์ Invoke URL
14. นำ URL ไปใส่ใน HTML หรือ JavaScript เพื่อเชื่อมกับ Lambda

ขั้นตอนการสร้าง Amazon SageMaker AI

1. ค้นหาบริการ SageMaker จากหน้า AWS Console
2. ไปที่แท็บ Notebook instances และคลิก Create notebook instance
3. ตั้งชื่อ instance ว่า lostandfound
4. ในหัวข้อ Notebook instance type ให้เลือก ml.t3.medium
5. ในหัวข้อ IAM role ให้เลือก LabRole
6. ตั้งค่า VPC เป็น Default
7. เลือก Subnet เป็น us-east-1a
8. เลือก Security group เป็นกลุ่มที่ชื่อ missing
9. จากนั้นคลิก Create notebook instance

ขั้นตอนการเชื่อมต่อกับ AWS Lambda

10. เปิดบริการ AWS Lambda แล้วเลือกแท็บ Functions
11. คลิกเข้าไปที่ฟังก์ชัน lostandfound
12. ในหน้าฟังก์ชัน เลือก Add trigger
13. เลือกบริการ S3 และเลือก Bucket ชื่อ lostfound123
14. ตีถูกยืนยัน I acknowledge that using the same S3 bucket... เพื่อยอมรับความเสี่ยง
15. คลิก Add เพื่อเพิ่ม trigger
16. เพิ่ม trigger อีกหนึ่งรายการ โดยเลือก API Gateway
17. เลือกตัวเลือก Use existing API

18. เลือก API ที่สร้างไว้จากรายการ Existing API และกด Add
19. จากนั้นไปที่ส่วน Code ได้หัวข้อ Code source เพื่อเพิ่มโค้ดในการดึงค่าจากบริการต่าง ๆ
20. กด Deploy เพื่อบันทึกและใช้งานโค้ด
21. ไปที่แท็บ Test และกำหนด Event JSON สำหรับการทดสอบ แล้วคลิก Save
22. กลับไปที่แท็บ Code แล้วคลิก Test เพื่อทดสอบการทำงานของระบบ

ผลการทดสอบการทำงานของระบบโซลูชัน (End-2-End)

Scenario 1: ผู้ใช้งานทั่วไปแจ้งข้อมูลคนหาย (User Journey)

เป้าหมาย: เพื่อทดสอบการทำงานของระบบ Lost & Found ตั้งแต่ขั้นตอนที่ผู้ใช้งานทั่วไปทำการแจ้งข้อมูลผู้สูญหายครบถ้วน จนถึงกรณีที่ระบบปัญญาประดิษฐ์ (AI) ทำการประเมินระดับความเสี่ยงของผู้สูญหายโดยอัตโนมัติ

ผู้มีส่วนเกี่ยวข้อง: ผู้ใช้งาน (User), ระบบ (System), ระบบ AI (AI System)

ลำดับขั้นตอนการทดสอบ (Detailed Flow):

1. เข้าสู่เว็บไซต์:

- ผู้ใช้งานเข้าสู่เว็บไซต์ Lost & Found
- ระบบจะแสดงหน้าแรกของเว็บไซต์

2. นำทางไปยังฟังก์ชันแจ้งข้อมูล:

- ผู้ใช้งาน คลิกที่เมนู "แจ้งข้อมูลคนหาย" ที่ปรากฏบนหน้าจอ
- ระบบจะนำทางผู้ใช้งานไปยังหน้าฟอร์มสำหรับกรอกข้อมูลการแจ้งคนหาย

3. การจัดการข้อมูลไม่ครบถ้วน (Validation Test):

- ผู้ใช้งานกรอกข้อมูลไม่ครบถ้วนในช่องที่กำหนด
- ผู้ใช้งาน คลิกปุ่ม "ยืนยัน"
- ระบบจะแสดงข้อความเตือน: "กรุณากรอกข้อมูลให้ครบทุกช่อง" ด้วยข้อความสีแดง เพื่อแจ้งให้ผู้ใช้งานทราบว่าข้อมูลยังไม่สมบูรณ์

4. การกรอกข้อมูลให้ครบถ้วนและถูกต้อง:

- ผู้ใช้งานกรอกข้อมูลของผู้สูญหายให้ครบถ้วน ตามรายการที่กำหนดในฟอร์ม:
 - ชื่อ-นามสกุล
 - ชื่อเล่น
 - อายุ
 - วันเกิด
 - คำหยา (เช่น รอยสัก, แผลเป็น)
 - รายละเอียดเกี่ยวกับผู้สูญหาย (ลักษณะนิสัย, สถานที่หาย, เหตุการณ์ก่อนเกิดเหตุ)
 - อีเมลผู้แจ้งผู้สูญหาย
 - เบอร์โทรผู้แจ้งผู้สูญหายติดต่อ

- ผู้ใช้งานอัปโหลดรูปภาพของผู้สูญหาย (จำนวน 1 รูปภาพ)
- 5. การยืนยันข้อมูลและการส่งเรื่อง:
 - ผู้ใช้งานคลิกปุ่ม "ยืนยัน" เพื่อส่งข้อมูลการแจ้งคนหาย
- 6. การประเมินความเสี่ยงโดย AI:
 - ระบบจะส่งข้อมูลที่ได้รับไปยังระบบ AI เพื่อทำการวิเคราะห์และประเมินระดับความเสี่ยงเบื้องต้นของผู้สูญหายโดยอัตโนมัติ
 - AI จะทำการจำแนกระดับความเสี่ยงเป็น "เสี่ยงสูง" / "เสี่ยงกลาง" / "เสี่ยงต่ำ" ตามเกณฑ์ที่กำหนด
- 7. การเผยแพร่ข้อมูล:
 - หลังจาก AI ประเมินความเสี่ยงและข้อมูลถูกบันทึกในฐานข้อมูล ระบบจะแสดงข้อมูลของผู้สูญหายที่แจ้งไปในหน้า "รวมประกาศหาคนหาย" เพื่อให้สาธารณะสามารถเข้าถึงและช่วยในการค้นหาได้

ผลลัพธ์ที่คาดหวัง:

- ข้อมูลถูกบันทึกสำเร็จ: ข้อมูลการแจ้งคนหายทั้งหมดที่ผู้ใช้งานกรอกไว้ จะต้องถูกบันทึกเข้าสู่ฐานข้อมูลของระบบอย่างถูกต้องสมบูรณ์
- AI ให้ระดับความเสี่ยงเบื้องต้น: ระบบ AI จะต้องประเมินและกำหนดระดับความเสี่ยง (สูง/กลาง/ต่ำ) สำหรับเคสผู้สูญหายที่แจ้งเข้ามา และแสดงผลลัพธ์นั้นในระบบ
- ข้อมูลแสดงในรายการคนหาย: ข้อมูลของผู้สูญหายที่แจ้งไป จะต้องปรากฏและสามารถมองเห็นได้ในส่วน "รวมประกาศหาคนหาย" บนเว็บไซต์

Scenario 2: Admin เข้าสู่ระบบและจัดการและปรับค่าการประเมินของ AI

เป้าหมาย: เพื่อทดสอบการทำงานของระบบแอดมินในการเข้าสู่ระบบ, การค้นหาข้อมูลเคสผู้สูญหาย, และการใช้สิทธิ์เพื่อเปลี่ยนแปลงผลการประเมินระดับความเสี่ยงจาก AI รวมถึงการจัดการสถานะของเคสโดยผู้ดูแลระบบ (Admin)

ผู้มีส่วนเกี่ยวข้อง: ผู้ดูแลระบบ (Admin), ระบบ (System), ระบบ AI (AI System)

ลำดับขั้นตอนการทดสอบ (Detailed Flow):

1. ผู้ใช้งานเข้าสู่เว็บไซต์:
 - ผู้ใช้งานเข้าสู่เว็บไซต์ Lost & Found
 - ระบบจะแสดงหน้าแรกของเว็บไซต์
2. การจัดการข้อมูลเข้าสู่ระบบไม่ถูกต้อง (Validation Test):
 - ผู้ดูแลระบบ กรอก Username หรือ Password ผิด: ระบบจะแสดงข้อความเตือน: "ผู้ใช้หรือรหัสผ่านไม่ถูกต้อง"
 - ผู้ดูแลระบบ ป้อนช่อง Username หรือ Password ว่างไว้: ระบบจะแสดงข้อความเตือน: "กรุณากรอกข้อมูลให้ครบ"
3. เข้าสู่ระบบแอดมินสำเร็จ:
 - ผู้ดูแลระบบ กรอก Username และ Password ที่ถูกต้อง
 - ผู้ดูแลระบบ คลิกปุ่ม "เข้าสู่ระบบ"
 - ระบบจะนำผู้ดูแลระบบเข้าสู่หน้าหลักของ Admin
4. การค้นหาข้อมูลคนหายที่ต้องการจัดการ:
 - ผู้ดูแลระบบ คลิกเมนู "ค้นหาข้อมูลคนหาย"
 - ในช่องค้นหา ผู้ดูแลระบบกรอกชื่อ "นางสาวลดา"
 - ระบบจะแสดงผลการค้นหาที่ตรงกัน เช่น "นางสาวลดา ไม้งาม"
5. การเข้าถึงรายละเอียดเคส:
 - ผู้ดูแลระบบ คลิกที่ชื่อ "นางสาวลดา ไม้งาม" ในรายการผลลัพธ์
 - ระบบจะ แสดงหน้าจอรายละเอียดเคส "นางสาวลดา ไม้งาม" ในมุมมองสำหรับ Admin ซึ่ง จะแสดงข้อมูลทั้งหมด รวมถึงผลการประเมินความเสี่ยงจาก AI
6. การตรวจสอบและแก้ไขระดับความเสี่ยงของ AI:
 - ผู้ดูแลระบบ ตรวจสอบผลการประเมินของ AI ซึ่งแสดงว่า: AI ให้ "เสี่ยงต่ำ"
 - ผู้ดูแลระบบ เห็นว่าควรปรับระดับความเสี่ยงเป็น "เสี่ยงกลาง"
 - ผู้ดูแลระบบ คลิกที่ Dropdown ที่แสดง "ความเสี่ยง: ต่ำ"
 - ผู้ดูแลระบบ เลือกเปลี่ยนเป็น "ความเสี่ยง: กลาง" จากรายการตัวเลือก
7. การจัดการสถานะของเคส:
 - ผู้ดูแลระบบ คลิกที่ Dropdown ที่แสดง "สถานะ: รับเรื่องแล้ว"
 - ผู้ดูแลระบบ เลือกเปลี่ยนเป็น "อยู่ในระหว่างค้นหา" จากรายการตัวเลือก
8. การบันทึกการเปลี่ยนแปลง:
 - ผู้ดูแลระบบ คลิกปุ่ม "บันทึก" เพื่อยืนยันการเปลี่ยนแปลง

- ระบบจะ แสดงข้อความยืนยันการบันทึกสำเร็จและบันทึกข้อมูลที่ถูกแก้ไขลงในฐานข้อมูล

9. การตรวจสอบความถูกต้องของการบันทึก:

- ผู้ดูแลระบบ ตรวจสอบหน้าจอรายละเอียดเคส เพื่อให้แน่ใจว่า ข้อมูลระดับความเสี่ยงและสถานะถูกอัปเดตเป็น "เสี่ยงกลาง" และ "อยู่ในระหว่างค้นหา" แล้ว
- ผู้ดูแลระบบ ดำเนินการ Logout ออกจากระบบ Admin
- ผู้ดูแลระบบ Login เข้าสู่ระบบ Admin อีกครั้ง ด้วย Username และ Password เดิม
- ผู้ดูแลระบบ กลับไปที่เคส "นางสาวลดา ไม้งาม" เพื่อตรวจสอบอีกครั้งว่าข้อมูลระดับความเสี่ยงและสถานะยังคงเป็นค่าล่าสุดที่บันทึกไว้ (นั่นคือ "เสี่ยงกลาง" และ "อยู่ในระหว่างค้นหา")

ผลลัพธ์ที่คาดหวัง:

- Admin สามารถแก้ไขการประเมินของ AI ได้: ผู้ดูแลระบบจะต้องสามารถเปลี่ยนแปลงระดับความเสี่ยงที่ AI ประเมินได้สำเร็จ
- สถานะเปลี่ยนจาก "รับเรื่องแล้ว" เป็น "อยู่ในระหว่างค้นหา": สถานะของเคสจะต้องถูกอัปเดตตามที่ผู้ดูแลระบบกำหนด
- ข้อมูลถูกบันทึกถาวร: การเปลี่ยนแปลงทั้งหมดที่ผู้ดูแลระบบทำ จะต้องถูกบันทึกในฐานข้อมูล
- ระบบแสดงระดับความเสี่ยงใหม่และสถานะใหม่ที่ Admin กำหนด: หน้าจอแสดงรายละเอียดเคสและส่วนอื่นๆ ที่เกี่ยวข้อง จะต้องสะท้อนระดับความเสี่ยงและสถานะที่ผู้ดูแลระบบเป็นผู้กำหนดล่าสุด

Scenario 3: การตรวจสอบผลกระทบหลังการแก้ไขของ Admin

เป้าหมาย: เพื่อยืนยันว่าการเปลี่ยนแปลงข้อมูลที่ผู้ดูแลระบบ (Admin) ทำการแก้ไข ได้รับการอัปเดตและแสดงผลอย่างถูกต้องในมุมมองของผู้ใช้งานทั่วไป (User)

ผู้มีส่วนเกี่ยวข้อง: ผู้ใช้งาน (User), ระบบ (System), ผู้ดูแลระบบ (Admin)

ลำดับขั้นตอนการทดสอบ (Detailed Flow):

1. ผู้ใช้งานเข้าสู่หน้าค้นหาข้อมูล:
 - ผู้ใช้งานทั่วไปเข้าสู่เว็บไซต์ Lost & Found
 - เข้าสู่หน้า “ค้นหาข้อมูลคนหาย”

2. การค้นหาเคสที่ถูกแก้ไข:

- ผู้ใช้งาน ใช้ฟังก์ชันค้นหาเพื่อค้นหาชื่อ "นางสาวลดา ไม้งาม"
- ระบบจะแสดงผลการค้นหาที่รวมถึง "นางสาวลดา ไม้งาม"

3. การตรวจสอบข้อมูลที่แสดงผลเบื้องต้น:

- ผู้ใช้งาน ตรวจสอบข้อมูลเบื้องต้นที่แสดงบนหน้ารวมประกาศ:
 - ตรวจสอบว่า ระดับความเสี่ยงแสดงเป็น "กลาง" แทนที่จะเป็น "ต่ำ" (ซึ่งเป็นค่าเริ่มต้นจาก AI)
 - ตรวจสอบว่า สถานะแสดงเป็น "อยู่ในระหว่างค้นหา" แทนที่จะเป็น "รับเรื่องแล้ว"

4. การทดสอบฟิลเตอร์ (Filter) ตามข้อมูลที่อัปเดต:

- ผู้ใช้งาน คลิกตัวเลือกฟิลเตอร์ "บุคคลที่มีความเสี่ยงต่ำ"
 - ระบบ ไม่ควรแสดง "นางสาวลดา ไม้งาม" ในรายการผลลัพธ์
- ผู้ใช้งาน คลิกตัวเลือกฟิลเตอร์ "บุคคลที่มีความเสี่ยงปานกลาง"
 - ระบบ ควรแสดง "นางสาวลดา ไม้งาม" ในรายการผลลัพธ์

5. การตรวจสอบรายละเอียดเคสหลังการอัปเดต:

- ผู้ใช้งาน คลิกที่ชื่อ "นางสาวลดา ไม้งาม" เพื่อเข้าสู่หน้าแสดงรายละเอียดข้อมูลผู้สูญหาย
- ผู้ใช้งาน ตรวจสอบข้อมูลล่าสุดที่แสดงบนหน้ารายละเอียด:
 - ตรวจสอบว่าระดับความเสี่ยงแสดงเป็น "กลาง"
 - ตรวจสอบว่าสถานะแสดงเป็น "อยู่ในระหว่างค้นหา"

ผลลัพธ์ที่คาดหวัง:

- ข้อมูลอัปเดตแสดงผลถูกต้องในฝั่ง User: การเปลี่ยนแปลงทั้งหมดที่ Admin ได้ทำการแก้ไข (ระดับความเสี่ยงและสถานะ) จะต้องปรากฏและแสดงผลอย่างถูกต้องในหน้า User Interface
- ฟิลเตอร์ทำงานตามข้อมูลใหม่: ระบบคัดกรอง (Filter) จะต้องประมวลผลและแสดงผลลัพธ์ตามข้อมูลล่าสุดที่ได้รับการอัปเดต ไม่ใช่ข้อมูลเดิม
- ไม่มี Cache หรือข้อมูลเก่าค้างอยู่: การอัปเดตข้อมูลต้องมีผลทันที และระบบจะต้องไม่แสดงข้อมูลเก่าที่ถูกแก้ไขไว้ หรือข้อมูลที่ไม่สอดคล้องกัน
- ข้อมูลสอดคล้องกันระหว่าง Admin และ User Interface: ข้อมูลเดียวกัน (เช่น ระดับความเสี่ยงและสถานะ) จะต้องแสดงผลตรงกันทั้งในส่วนจัดการของ Admin และในส่วนที่ผู้ใช้งานทั่วไปมองเห็น

บทวิเคราะห์ข้อดีและข้อเสียของระบบโดยอิงตาม AWS Well-Architected Framework

Operational excellence questions

Organization

How do you determine what your priorities are?

มีการนัดพูดคุยกันภายในทีมเพื่อให้สมาชิกทุกคนมีความเข้าใจเกี่ยวกับระบบไปในทิศทางเดียวกัน จากนั้นนำข้อสรุปที่ได้มาทำเอกสารแล้วจึงทำงานตามลำดับที่ตกลงกันได้

How do you structure your organization to support your business outcomes?

แบ่งงานเป็นส่วนย่อย ๆ แล้วแจกแจงให้สมาชิกแต่ละคนตามความถนัดและเหมาะสม เพื่อให้ดำเนินการได้รวดเร็วไม่เกิ่นการทำงานชนกัน

How does your organizational culture support your business outcomes?

เนื่องจากสมาชิกทุกคนมีส่วนร่วมในการออกความเห็น ทั้งในส่วนของการออกแบบระบบ และการแบ่งหน้าที่รับผิดชอบ จึงมั่นใจได้ว่าจะได้รับงานที่มีคุณภาพจากทุกฝ่าย

Prepare

How do you design your work load so that you can understand its state?

ออกแบบระบบให้ดูสถานะได้ง่าย เช่น ใช้ CloudWatch เช็กว่าแต่ละส่วนยังทำงานอยู่ไหม

How do you reduce defects, ease remediation, and improve flow into production?

ลดข้อผิดพลาดด้วยการทดสอบบ่อย ๆ และจัดระบบการส่งขึ้น production ที่ชัดเจน

How do you mitigate deployment risks?

ทดสอบระบบด้วยตนเองก่อนปล่อยจริง โดยเช็กฟังก์ชันหลักให้แน่ใจว่าใช้งานได้ และเก็บโค้ดเวอร์ชันก่อนหน้าไว้เพื่อเกิดปัญหา

How do you know that you are ready to support a workload?

ก่อนจะให้ระบบไปรันจริง เราจะใช้ CloudWatch log ตรวจสอบการทำงานของระบบก่อน

Operate

How do you understand the health of your workload?

ใช้ log ไฟล์ หรือ error message เพื่อดูว่าทำงานปกติ หรือทดสอบเรียก API แล้วดูว่าตอบกลับถูกต้องหรือไม่

How do you understand the health of your operations?

ดูค่า latency, error rate และเช็กสุขภาพของระบบทุกส่วน

How do you manage workload and operations events?

มีการเข้าตรวจสอบระบบเป็นระยะ หากพบปัญหาจะรีบเข้าเช็ก CloudWatch log ของ Lambda และทำการแก้ไข

Evolve

How do you evolve operations?

ดูจาก CloudWatch log และ feedback ของผู้ใช้ แล้วนำมาปรับปรุงระบบ ให้เร็วขึ้นหรือใช้งานง่ายขึ้น

Security questions

Security

How do you securely operate your workload?

กำหนดสิทธิ์การเข้าถึงเฉพาะที่จำเป็น เช่น Lambda เข้าถึงเฉพาะบางตารางในฐานข้อมูล และใช้ HTTPS ในการส่งข้อมูล แยกข้อมูลสำคัญไว้ในพื้นที่ปลอดภัย เช่น S3 ที่เปิดเฉพาะบาง role

How do you manage identities for people and machines?

ใช้ Security Group และ IAM Lab Rule

How do you manage permissions for people and machines?

เรากำหนดสิทธิ์แบบจำกัดเฉพาะที่จำเป็น ให้เข้าถึงเฉพาะ resource ที่ต้องใช้

Detection

How do you detect and investigate security events?

มีการตรวจสอบ CloudWatch log อย่างสม่ำเสมอ และใช้ระบบแจ้งเตือนของแอปหรือ service ที่ใช้อยู่

How do you protect your network resources?

ใช้ VPC, Security Group กำหนดว่าใครมีสิทธิ์ในทรัพยากรบ้าง

How do you protect your compute resources?

จำกัดสิทธิ์การเข้าถึงด้วย IAM Role และใช้ Security Group เพื่อไม่ให้เข้าถึงทรัพยากรอื่นเข้าถึงโดยไม่จำเป็น

Data Protection

How do you classify your data?

แยกข้อมูลตามประเภท เช่น รูปภาพและโมเดล AI ใน S3, SQL code ใน RDS เพื่อจัดเก็บและป้องกันอย่างเหมาะสม

How do you protect your data at rest?

ใช้การเข้ารหัส (encryption) แบบ Enable Default Encryption สำหรับ S3, Aurora และ RDS

How do you protect your data in transit?

ใช้ HTTPS (TLS) เพื่อเข้ารหัสข้อมูลขณะส่งระหว่างระบบ เช่น ระหว่าง Frontend กับ API Gateway และจาก Lambda ไปยัง RDS

Incident response

How do you anticipate, respond to, and recover from incidents?

เมื่อเกิดข้อผิดพลาด เช่น Lambda มี error จะใช้ cloudwatch วิเคราะห์หาต้นเหตุของปัญหา และทำการ rollback กลับไปยังงาน (version) ก่อนหน้า

Reliability question

Foundations

How do you manage service quotas and constraints?

ตรวจสอบการใช้ทรัพยากรผ่าน AWS Console และวางแผนใช้งานไม่ให้เกินโควตาที่กำหนด

How do you plan your network topology?

ใช้ VPC เพื่อแยกส่วนเครือข่าย แบ่งเป็น public subnet สำหรับ API Gateway และ private subnet สำหรับ Lambda กับ RDS

Workload architecture

How do you design your workload service architecture?

: แยกการประมวลผล (Lambda), การจัดเก็บ (S3), ฐานข้อมูล(RDS) และระบบโมเดล ML (Sagemaker) ออกเป็นแต่ละหน้าที่อย่างชัดเจน และมีการเชื่อมต่อกับ API Gateway เพื่อดึงและรับข้อมูลจากผู้ใช้งาน

- How do you design interactions in a distributed system to prevent failure?

แยกส่วนการทำงานตามหน้าที่ เช่น API Gateway รับคำขอ, Lambda ประมวลผล, RDS เก็บข้อมูล, S3 เก็บไฟล์ และ SageMaker สำหรับประมวลผล ML เพื่อให้ระบบยืดหยุ่นและดูแลง่าย

- How do you design interactions in a distributed system to mitigate or withstand failures?

ออกแบบให้แต่ละบริการแยกกันทำงาน เช่น Lambda แบบ stateless

Change management

- How do you monitor workload resources?

: ใช้ CloudWatch ตรวจสอบ log และสถานะของ Lambda, API

- How do you design your workload to adapt to changes in demand?

ใช้ Lambda ที่สามารถ scale ตามจำนวนคำขอ และเลือกบริการที่ยืดหยุ่น เช่น RDS ที่ปรับขนาด instance ได้เมื่อต้องการ

- How do you implement change?

ทดสอบการเปลี่ยนแปลงในสภาพแวดล้อมจำลองก่อน แล้วค่อยนำขึ้นระบบจริงแบบทีละขั้น เพื่อลดความเสี่ยง

Failure management

- How do you back up data?

สำรองข้อมูลจาก RDS แบบอัตโนมัติ และเก็บไฟล์ใน S3 ซึ่งมีการทำสำเนาหลายชุดภายใน AWS อยู่แล้ว

- How do you use fault isolation to protect your workload?

แยกบริการออกเป็นส่วน ๆ เช่น Lambda, RDS, และ S3 ทำงานแยกจากกันหากส่วนใดล้ม จะไม่กระทบระบบทั้งหมด

- How do you design your workload to withstand component failures?

เป็น **Serverless** และ **Managed Service** ของ AWS (Lambda, S3, API Gateway, SageMaker) ที่มีความทนทานสูงและ Scale อัตโนมัติเป็นหลัก พร้อมทำให้ **Lambda Functions** เป็น **Stateless** แยกส่วนประกอบจำกัดผลกระทบจากการล้มเหลว

- How do you test reliability?

ทดลองใช้จริงเพื่อดูการทำงานของระบบทั้งหมด

- How do you plan for disaster recovery?

ใช้ Backup & Restore จาก RDS Snapshots และ S3 Versioning

Performance efficiency questions

Selection

- How do you select the best performing architecture?

เลือกใช้สถาปัตยกรรมแบบแยกส่วน เช่น Lambda สำหรับประมวลผล, RDS สำหรับฐานข้อมูล และ S3 สำหรับเก็บไฟล์

- How do you select your compute solution?

เลือก AWS Lambda เป็น solution compute หลัก เพราะ Scale อัตโนมัติตามความต้องการ, จ่ายตามการใช้งานจริง, และไม่ต้องจัดการ Server

- How do you select your storage solution?

ใช้ S3 สำหรับจัดเก็บรูปภาพและไฟล์ เนื่องจากรองรับได้เยอะ ราคาถูกและเข้าถึงได้เร็ว

- How do you select your database solution?

เลือก Amazon RDS เป็น solution database เพราะข้อมูลมี Schema ชัดเจนและมีความสัมพันธ์, ต้องการ Transactional Consistency (ACID) และเป็น Managed Service ที่ลดภาระการดูแล

- How do you configure your networking solution?

วางระบบให้อยู่ใน Region เดียวกัน ใช้ API Gateway เชื่อม Frontend กับ Backend อย่างปลอดภัย และใช้ Private Network ภายใน AWS

Review

- How do you evolve your workload to take advantage of new releases?

ติดตามข่าวสารจาก AWS เช่น AWS News, Blog และ Release Note

แล้วนำมาวิเคราะห์ว่าบริการใหม่ใดเหมาะกับระบบและ IAM ของตนเอง เพื่อนำไปประยุกต์ใช้ใน อนาคต

Monitoring

- How do you monitor your resources to ensure they are performing?

ตรวจสอบการทำงานของบริการหลัก เช่น Lambda, RDS, S3 ผ่านการดู log และ performance metrics เช่น response time, error rate เพื่อให้แน่ใจว่าระบบยังทำงานได้ดี

Tradeoffs

- How do you use tradeoffs to improve performance?

ใช้ Lambda และ S3 แทน EC2 ในงานบางประเภท, ใช้ RDS แทน Aurora เพื่อลดต้นทุน

Cost optimization

Practice cloud financial management

- How do you implement cloud financial management?

มีการกำหนดงบประมาณและวางแผนการใช้บริการคลาวด์ล่วงหน้าอย่างเหมาะสม

- How do you evaluate cost when you select services?

พิจารณาค่าใช้จ่ายและลักษณะงานก่อนเลือกใช้บริการ

- How do you meet cost targets when you select resource type, size, and number?

เลือกใช้บริการที่คิดค่าบริการตามการใช้งานจริง เช่น Lambda และใช้ RDS

ขนาดเล็กพอเหมาะกับโหลดที่มี เพื่อไม่ให้สิ้นเปลือง

รวมถึงกำหนดการใช้งานให้ตรงกับความเป็นจริง

- How do you use pricing models to reduce cost?

: ใช้ Lambda แบบ pay-per-request, RDS แทน Aurora เพื่อลดต้นทุน

- How do you plan for data transfer changes?

: วางระบบทั้งหมดใน Region เดียวกัน เพื่อความสะดวกสบายและลดค่าใช้จ่าย

Expenditure and usage awareness

- How do you govern usage?

: ใช้ Security group จำกัดสิทธิ์ในการใช้งานทรัพยากร ป้องกันการใช้เกินจำเป็น

- How do you monitor usage and cost?

: ใช้ CloudWatch เพื่อติดตามค่าใช้จ่าย

- How do you decommission resources?

: หยุดบริการ AWS อย่างอื่นเมื่อไม่ได้ใช้งาน

Manage demand and supply resources

- How do you manage demand and supply resources?

ใช้ Lambda ที่ scale ตามคำขอ และ Auto Scaling เพื่อปรับตามปริมาณทรัพยากรที่ใช้งาน เพื่อไม่ให้ค่าใช้จ่ายเกินจำเป็น

Optimize over time

- How do you evaluate new services?

ศึกษาคุณสมบัติของบริการใหม่จาก AWS แล้วพิจารณาว่าตอบโจทย์ระบบเราหรือไม่ ถ้าเหมาะสมก็จะทดลองใช้งานในระบบทดสอบก่อน

บทสรุปข้อดี-ข้อเสียของระบบ Lost and Found (AWS Well-Architected Framework)

ข้อดี:

- **Operational Excellence:**

- ทีมงานสื่อสารและเข้าใจตรงกัน, แบ่งงานชัดเจน, มีส่วนร่วมในการออกแบบ
- ใช้ CloudWatch/Log ตรวจสอบสถานะระบบง่าย
- มีการทดสอบและจัดระบบส่งขึ้น Production, เก็บโค้ดเวอร์ชันก่อนหน้า
- ปรับปรุงระบบต่อเนื่องจาก Log และ Feedback ผู้ใช้

- **Security:**

- กำหนดสิทธิ์แบบ **Least Privilege** (Lambda เข้าถึงเฉพาะที่จำเป็น) ด้วย **IAM Role** และ **Security Group**
- เข้ารหัสข้อมูล ทั้ง **Data at Rest** (S3, RDS) และ **Data in Transit** (HTTPS/TLS)
- แยกข้อมูลตามประเภทเพื่อป้องกันที่เหมาะสม
- ตรวจสอบ **CloudWatch Log** และใช้ระบบแจ้งเตือนเพื่อตรวจจับความผิดปกติ
- ใช้ **VPC** เพื่อป้องกันเครือข่าย

- **Reliability:**

- ใช้ **Serverless & Managed Services** (Lambda, S3, API Gateway) ที่ Scale อัตโนมัติและทนทานในตัว
- Lambda เป็น Stateless ลดผลกระทบเมื่อเกิดปัญหา
- แยกการทำงานของระบบจำกัดผลกระทบความล้มเหลว
- มีการ สำรองข้อมูล RDS และ S3
- ระบบปรับตัวตาม Demand ได้ด้วย Lambda Scaling
- ทดสอบในสภาพแวดล้อมจำลองก่อน Deploy

- **Performance Efficiency:**

- เลือก Serverless Architecture (Lambda, S3) ที่ Scale ได้ดี
- **Lambda** เป็น Compute Solution ที่ยืดหยุ่นและประหยัด
- **S3** เป็น Storage ที่รองรับไฟล์จำนวนมากและราคาถูก
- **RDS** เป็นฐานข้อมูลที่มีคุณภาพและประหยัดต้นทุน
- เครือข่ายใน Region เดียวกัน, ใช้ API Gateway และ Private Network ลด Latency

- **Cost Optimization:**

- กำหนดงบประมาณ และ พิจารณาค่าใช้จ่าย ก่อนเลือกบริการ
- ใช้ Pay-per-Use Service (Lambda) และ RDS ขนาดเหมาะสม
- วางระบบใน Region เดียวกัน ลดค่า Data Transfer
- ใช้ Security Group ควบคุมการใช้งาน, CloudWatch ควบคุมค่าใช้จ่าย
- หยุดบริการที่ไม่ใช้งาน, ประเมินบริการใหม่ๆ

ข้อเสีย:

- **Reliability:**

- **RDS เป็น Single-AZ:** หาก AZ ที่ RDS อยู่ล่ม ฐานข้อมูลและระบบจะหยุดทำงานทันที
- การทดสอบ **Reliability** ยังไม่ละเอียด: ไม่ได้มีการเก็บข้อมูลกราฟหรือคำนวณแบบเก็บข้อมูลวัดผล

- **Security:**

- **IAM Lab Rule:** ไม่มีสิทธิ์กำหนด IAM ได้เอง จึงอาจมีช่องโหว่หรือสิทธิ์ที่ไม่ได้รับอนุญาต

- **Operational Excellence:**
 - ไม่มีการ **Automated Testing**
 - การตรวจสอบความพร้อมของ Workload ก่อนรันจริงยังไม่ครอบคลุม Load Test.
 - การจัดการ Event และการแจ้งเตือนไม่ได้รัดกุมและชัดเจน
- **Performance Efficiency:**
 - การ Scaling ของ RDS ใน Single-AZ เป็นข้อจำกัดในการโหลด
- **Cost Optimization:**
 - การเลือก Single-AZ เพื่อลดต้นทุน แลกมาด้วยความเสี่ยงด้าน Reliability

URL

วิดีโอ Demo ระบบ

<https://drive.google.com/file/d/1jAbZL8qcfPBuAXs1Y3RInHHM7-0cs8oH/view?usp=sharing>

Link Github Code Frontend

<https://github.com/anuttamii/CS232.git>

สรุปผลการดำเนินโครงการ

ในการดำเนินงานการพัฒนาเว็บไซต์ Lost&Found เพื่อเป็นเว็บไซต์ในการรวบรวมการแจ้งข้อมูลผู้สูญหาย โดยมีการนำเอาเทคโนโลยี Cloud และบริการจาก AWS มีฟังก์ชันที่ช่วยให้มีการใช้งานง่ายและเหมาะสม โดยข้อมูลจะถูกจัดเก็บในฐานข้อมูลของ Amazon RDS และทำการจัดเก็บไฟล์รูปภาพไว้บน AWS S3 Bucket และยังสามารถนำเอา AWS Lambda เพื่อทำหน้าที่เป็นตัวกลางในการรับส่งข้อมูลจากหน้า Website ผ่านทาง Amazon API Gateway เพื่อส่งข้อมูลไปเก็บยังฐานข้อมูลของ Amazon RDS ตามที่ได้สร้างมาและแสดงผลได้อย่างเป็นระบบ

ซึ่งผลที่ได้จากการพัฒนาระบบนี้ ทำให้เว็บไซต์ Lost&Found สามารถได้ตามฟีเจอร์หลักๆ ได้ เช่น ผู้ใช้สามารถแจ้งข้อมูลผู้สูญหายได้ ผู้ใช้สามารถดูข้อมูลผู้สูญหายทั้งหมดได้ ผู้ใช้สามารถดูข้อมูลรายละเอียดคนหายรายบุคคลได้ สามารถทราบความเสี่ยงและสถานะของบุคคลสูญหายได้

โดยปัญหาและอุปสรรคที่พบเจอระหว่างการทำงาน พบว่า เนื่องจากใน Learner Lab มีการจำกัดการเข้าถึงบริการต่างๆอยู่ จึงทำให้บางบริการที่ได้วางแผนเอาไว้ไม่สามารถทำได้จริง เนื่องจากอยู่นอกเหนือจากบริการที่ Learner Lab ให้มา และสิทธิ์ในการเข้าถึง ที่มีสิทธิ์เพียงแค่ LabRole รวมถึงมีอุปสรรคในด้านของเวลาที่จำกัดในการพัฒนาระบบนี้ขึ้นมา

โดยมีแนวทางการแก้ไขปัญหา ดังนี้ ในบางบริการซึ่งไม่มีใน Learner Lab ได้ทำการเปลี่ยนบริการเป็นบริการที่รองรับใน Learner Lab เช่น จากข้อเสนอโปรเจกต์ได้กำหนดไว้ว่าต้องการใช้บริการ AWS Comprehend จึงทำการเปลี่ยนเป็น Amazon SageMaker AI และปรับให้ใช้ได้สิทธิ์ที่ได้รับมา และในการแก้ไขปัญหาเรื่องเวลาที่มีจำกัด จึงได้ทำการลดประสิทธิภาพในการทำเว็บไซต์ในบางส่วน เพื่อให้เหมาะสมกับเวลาที่จำกัดมากขึ้น

แนวทางพัฒนาต่อในอนาคตเกี่ยวกับการพัฒนาการใช้ Machine Learning ให้มีประสิทธิภาพมากขึ้น เติบโตมากยิ่งขึ้น การนำเอาไปพัฒนาเรื่องความใช้งานง่ายของเว็บไซต์มากยิ่งขึ้น รวมถึงการเพิ่มฟีเจอร์ให้มีฟีเจอร์ที่เป็นประโยชน์ต่อการรวบรวมข้อมูลผู้สูญหายมากยิ่งขึ้น