



## เอกสารโครงการ

ระบบบริหารจัดการกิจกรรมและติดตามการพัฒนาทักษะของนักศึกษา (ANTivity)

### จัดทำโดย

กลุ่มที่ 3

6609611808	ฉัพปนัย	เกิดสมจิตร์
6609611816	ชนกันต์	ทองรอง
6609611907	ณัฐชนน	สาระสังข์
6609611915	ณัฐดนัย	พันธุ์เสนา
6609611972	ธนาภูมิ	ผลเมืองดี
6609611980	ธนช	เกิดพิพิร্য
6609612160	ภัทรภูมิ	กิงซ์ย
6609612269	สุภานัย	จิรกาลกุล

### เสนอ

ผศ.ดร.วนิดา พฤทธิวิทยา

ผศ.ดร.ประภาพร รัตนธิรัง

รายงานนี้เป็นส่วนหนึ่งของวิชา คพ.232 เทคโนโลยีกลุ่มเมฆเบื้องต้น

(INTRODUCTION TO CLOUD COMPUTING TECHNOLOGY) ภาคการศึกษาที่ 2 ปีการศึกษา 2567

คณะวิทยาศาสตร์และเทคโนโลยี สาขาวิชาวิทยาการคอมพิวเตอร์ (คอมพิวเตอร์และวิทยาการสารสนเทศ)

มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต

## บทคัดย่อ

โครงการนี้มีวัตถุประสงค์เพื่อพัฒนาระบบ ANTivity ซึ่งเป็นระบบสารสนเทศเพื่อสนับสนุนการติดตามและพัฒนาทักษะของนักศึกษาให้สอดคล้องกับกรอบสมรรถนะตามหลักสูตรการศึกษา โดยเน้นการเรียนรู้ผ่านการเข้าร่วมกิจกรรมที่จัดขึ้นนอกห้องเรียน ทั้งในรูปแบบของกิจกรรมเสริมทักษะทางวิชาชีพ (Hard Skills) และทักษะด้านพฤติกรรม (Soft Skills) ระบบรองรับการใช้งานในสามบทบาทหลัก ได้แก่ นักศึกษา อาจารย์ที่ปรึกษา และเจ้าหน้าที่ผู้จัดกิจกรรม (Staff) ซึ่งมีบทบาทร่วมกันในการส่งเสริม พัฒนา และติดตามความก้าวหน้าทางทักษะของนักศึกษา

ในระบบนี้ นักศึกษามีความสามารถค้นหา สมัคร และเข้าร่วมกิจกรรมที่เกี่ยวข้อง พร้อมทั้งตรวจสอบพัฒนาการของตนเองได้อย่างต่อเนื่อง อาจารย์ที่ปรึกษามีความสามารถติดตามความก้าวหน้าทางทักษะของนักศึกษาในความดูแล และประเมินความครบถ้วนของสมรรถนะตามเกณฑ์ที่หลักสูตรกำหนด ส่วนเจ้าหน้าที่ผู้จัดกิจกรรมสามารถจัดการกิจกรรม วิเคราะห์ความต้องการทักษะในภาพรวม อนุมัติการเข้าร่วมกิจกรรม ประเมินผลการเข้าร่วม และจัดการทรัพยากรได้อย่างมีประสิทธิภาพ โดยระบบมีฟังก์ชันสนับสนุนการวิเคราะห์ช่องว่างด้านทักษะของนักศึกษาแต่ละราย และของกลุ่มนักศึกษาในภาพรวม เพื่อวางแผนกิจกรรมในอนาคตอย่างตรงจุด

ระบบพัฒนาด้วย Next.js สำหรับส่วนแสดงผล ส่วน Backend ของโครงการพัฒนาด้วย Hono Framework ซึ่งทำงานบน Deno Runtime โดยโญาสต์ทั้งสองส่วนบน Amazon EC2 การจัดการผู้ใช้และการยืนยันตัวตนดำเนินการผ่าน Amazon Cognito ข้อมูลกิจกรรมและพัฒนาการทักษะถูกจัดเก็บในฐานข้อมูล Amazon RDS และไฟล์รูปภาพจัดเก็บไว้บน Amazon S3 ระบบมีการออกแบบให้สามารถปรับขยายได้ (Scalable) และรองรับการใช้งานจริงในสภาพแวดล้อมของสถาบันการศึกษา

## สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
สารบัญ	ข-ค
บทที่ 1: ข้อมูลเบื้องต้นของโครงการ	1-4
1.1 ชื่อระบบและแนวคิดเบื้องต้น	1
1.2 ที่มาและความสำคัญของโครงการ	1
1.3 วัตถุประสงค์ของระบบ	2
1.4 ปัญหาและความต้องการที่ระบบตอบโจทย์	3
1.5 ขอบเขตของโครงการ	4
บทที่ 2: การออกแบบระบบและการใช้งาน	5-7
2.1 บทบาทผู้ใช้งานและกรณีการใช้งาน (Use Case Scenarios)	5
2.2 ฟังก์ชันหลักของระบบและประโยชน์ที่ได้รับ	7
บทที่ 3: สถาปัตยกรรมระบบ	9-12
3.1 ภาพสถาปัตยกรรมระบบโดยรวม	9
3.2 รายละเอียดบริการ AWS ที่ใช้	10
3.3 การเชื่อมโยงและการทำงานร่วมกันของแต่ละองค์ประกอบ	12
3.4 ประโยชน์ของสถาปัตยกรรมนี้	12
บทที่ 4: ขั้นตอนการติดตั้งและตั้งค่าระบบ	13-19
4.1 การตั้งค่า AWS Cognito สำหรับการจัดการผู้ใช้	13
4.2 การตั้งค่า Amazon RDS และการออกแบบฐานข้อมูล	15
4.3 การตั้งค่า Amazon S3 สำหรับจัดเก็บไฟล์	17
4.4 การติดตั้งและติดตั้งแพลตฟอร์ม (Next.js + Hono บน EC2)	19
บทที่ 5: ผลการทดสอบระบบและการรับมือกับข้อผิดพลาด	21-28
5.1 รายงานผลการทดสอบแบบ End-to-End	21
5.2 กรณีที่ระบบทำงานปกติ	28
5.3 กรณีที่เกิดข้อผิดพลาดและระบบรับมือ	28

เรื่อง	หน้า
บทที่ 6: การวิเคราะห์ระบบด้วย AWS Well-Architected Framework	30-37
6.1 Operational Excellence (ประสิทธิภาพการปฏิบัติงาน)	30
6.2 Security (ความปลอดภัย)	31
6.3 Reliability (ความน่าเชื่อถือ)	33
6.4 Performance Efficiency (ประสิทธิภาพ)	34
6.5 Cost Optimization (การจัดการต้นทุน)	36
6.6 Sustainability (ความยั่งยืน)	37
บทที่ 7: สรุปและแนวทางพัฒนาต่อ	39-44
7.1 สรุปผลการดำเนินงานของโครงการ	39
7.2 ปัญหาและอุปสรรคที่พบ	40
7.3 แนวทางการพัฒนาต่ออยอดในอนาคต	42
7.4 ข้อเสนอแนะเพื่อการปรับปรุงระบบ	44

ภาคผนวก

# บทที่ 1

## ข้อมูลเบื้องต้นของโครงการ

### 1.1 ชื่อระบบและแนวคิดเบื้องต้น

ระบบที่พัฒนาขึ้นในโครงการนี้มีชื่อว่า “ANTivity” ซึ่งเป็นการผสมคำระหว่าง “Ant” ที่สื่อถึงความขยัน ความมีระเบียบ และการทำงานเป็นทีม กับคำว่า “Activity” ที่หมายถึงกิจกรรม โดยตั้งใจสื่อถึงการส่งเสริมให้นักศึกษาเข้าร่วมกิจกรรมต่าง ๆ เพื่อพัฒนาทักษะนักเรียนอย่างเป็นระบบ มีเป้าหมายหลักในการติดตามความก้าวหน้าและสนับสนุน การพัฒนาทักษะทั้งในด้าน Soft Skills และ Hard Skills ให้สอดคล้องกับผลลัพธ์การเรียนรู้ที่กำหนดไว้ในหลักสูตร การศึกษา

แนวคิดของระบบมุ่งเน้นการบูรณาการบทบาทของผู้ใช้ 3 กลุ่มหลัก ได้แก่

- นักศึกษา: ผู้ใช้ระบบเพื่อติดตามพัฒนาการด้านทักษะของตนเองจากการเข้าร่วมกิจกรรม
- อาจารย์ที่ปรึกษา: ผู้ใช้ระบบเพื่อติดตาม ประเมิน และให้คำแนะนำแก่นักศึกษาได้อย่างตรงจุด
- เจ้าหน้าที่หรือผู้จัดกิจกรรม: ผู้ใช้ระบบเพื่อติดตามแนวโน้มความนิยมของกิจกรรม วางแผนจัดกิจกรรม และบริหารข้อมูลการเข้าร่วม

ระบบ ANTivity ถูกออกแบบให้เป็นระบบศูนย์กลางในการเก็บรวบรวมข้อมูลกิจกรรมของนักศึกษาอย่างเป็นระบบ และแสดงผลข้อมูลพัฒนาการด้านทักษะผ่านแดชบอร์ด (Dashboard) และรายงานวิเคราะห์ เพื่อตอบโจทย์ทั้งเชิงบริหารจัดการกิจกรรมและเชิงสนับสนุนการเรียนรู้ตลอดระยะเวลาการศึกษา โดยอาศัยเทคโนโลยี Cloud Computing บนแพลตฟอร์ม Amazon Web Services (AWS) เพื่อให้ระบบมีความยืดหยุ่น ปรับขยายได้ และปลอดภัยในการใช้งาน

### 1.2 ที่มาและความสำคัญของโครงการ

ในระบบการศึกษาระดับอุดมศึกษายุคปัจจุบัน การพัฒนาทักษะของนักศึกษาไม่ได้จำกัดอยู่เพียงภาษาในห้องเรียน หากแต่ขยายออกไปสู่กิจกรรมเสริมหลักสูตรหลากหลายรูปแบบ เช่น กิจกรรมจิตอาสา การอบรมทักษะเฉพาะทาง การสัมมนาวิชาการ หรือการมีส่วนร่วมในกิจกรรมขององค์กรนักศึกษา โดยกิจกรรมเหล่านี้มีบทบาทสำคัญในการส่งเสริมทักษะด้านชีวิต (Soft Skills) เช่น ความเป็นผู้นำ การสื่อสาร และการทำงานเป็นทีม ตลอดจน ทักษะทางวิชาชีพ (Hard Skills) เช่น การเขียนโปรแกรม การวิเคราะห์ข้อมูล และการใช้เทคโนโลยี

อย่างไรก็ตาม ในสภาพแวดล้อมของมหาวิทยาลัยปัจจุบัน การจัดเก็บและติดตามข้อมูลกิจกรรมของนักศึกษามักดำเนินการแยกส่วนตามหน่วยงานต่าง ๆ เช่น ฝ่ายกิจการนักศึกษา ชุมชน หรือแต่ละคณะ ส่งผลให้นักศึกษา ไม่สามารถมองเห็นภาพรวมของกิจกรรมที่ตนเองเข้าร่วม หรือ วิเคราะห์ทักษะที่ได้รับอย่างเป็นระบบ ได้ ขณะเดียวกัน อาจารย์ที่ปรึกษา ก็ขาดข้อมูลเชิงสังเคราะห์ที่ช่วยให้สามารถติดตามและประเมินพัฒนาการของนักศึกษาแต่ละคนหรือทั้งกลุ่มได้อย่างมีประสิทธิภาพ ส่วน เจ้าหน้าที่หรือผู้จัดกิจกรรม ก็ไม่สามารถวิเคราะห์แนวโน้มความสนใจและผลลัพธ์ของกิจกรรมในภาพรวมได้อย่างชัดเจน

จากปัญหาดังกล่าว จึงเกิดแนวคิดในการพัฒนาโครงงานระบบ ANTivity เพื่อเป็นระบบกลางสำหรับรวมบันทึก และวิเคราะห์ข้อมูลกิจกรรมของนักศึกษา พร้อมเชื่อมโยงกิจกรรมเข้ากับทักษะที่ได้รับในรูปแบบที่เข้าใจง่าย โดยระบบถูกออกแบบให้รองรับการใช้งานของผู้มีส่วนเกี่ยวข้องหลัก ได้แก่

นักศึกษา: ใช้ระบบเพื่อติดตามพัฒนาการของตนเอง วิเคราะห์ทักษะที่ได้รับ และค้นหากิจกรรมที่เหมาะสม  
อาจารย์ที่ปรึกษา: ใช้ระบบเพื่อตรวจสอบความก้าวหน้าเชิงทักษะของนักศึกษา และให้คำแนะนำได้อย่างตรงจุดเจ้าหน้าที่ผู้จัดกิจกรรม: ใช้ระบบเพื่อจัดการกิจกรรม วิเคราะห์แนวโน้มความสนใจ และประเมินผลกิจกรรมที่จัดระบบ ANTivity จึงมีความสำคัญทั้งในเชิงบริหารจัดการกิจกรรมนักศึกษาอย่างเป็นระบบ และในเชิงส่งเสริมการเรียนรู้เชิงทักษะอย่างรอบด้าน ซึ่งสอดคล้องกับแนวทางการเรียนรู้ในศตวรรษที่ 21 และสนับสนุนการเตรียมความพร้อมของนักศึกษาในการเข้าสู่ตลาดแรงงานอย่างมีคุณภาพ

### 1.3 วัตถุประสงค์ของระบบ

โครงงานระบบ ANTivity มีวัตถุประสงค์หลักเพื่อพัฒนาระบบสารสนเทศกลางที่สนับสนุนการติดตาม วิเคราะห์ และส่งเสริมพัฒนาการทางทักษะของนักศึกษา โดยมีวัตถุประสงค์ย่อยดังต่อไปนี้:

1. เพื่อพัฒนาระบบที่สามารถสำรวจและแสดงผลประวัติการเข้าร่วมกิจกรรมของนักศึกษาในรูปแบบที่เข้าใจง่าย พร้อมเชื่อมโยงกับทักษะที่ได้รับจากแต่ละกิจกรรมอย่างเป็นระบบ
2. เพื่อออกแบบระบบวิเคราะห์ทักษะของนักศึกษา โดยสามารถแสดงผลความก้าวหน้าด้าน Soft Skills และ Hard Skills และทักษะที่ยังขาดตามกรอบสมรรถนะของหลักสูตร
3. เพื่อพัฒนาแดชบอร์ด (Dashboard) สำหรับอาจารย์ที่ปรึกษาในการติดตามและประเมินพัฒนาการเชิงทักษะของนักศึกษารายบุคคลและรายชั้นปี เพื่อประกอบการให้คำปรึกษาเชิงลึก
4. เพื่อสร้างระบบสนับสนุนการจัดกิจกรรมของเจ้าหน้าที่ โดยสามารถจัดการกิจกรรม ประเมินผู้เข้าร่วม อนุมัติการสมัคร วิเคราะห์แนวโน้มความสนใจ และประเมินผลกิจกรรมเชิงสถิติได้อย่างมีประสิทธิภาพ
5. เพื่อส่งเสริมการเรียนรู้และการพัฒนาทักษะของนักศึกษาอย่างต่อเนื่อง ผ่านการใช้เทคโนโลยีสารสนเทศที่มีความสามารถในการปรับขยาย ปลอดภัย และใช้งานง่าย บนแพลตฟอร์ม AWS Cloud

## 1.4 ปัญหาและความต้องการที่ระบบตอบโจทย์

ในการพัฒนาทักษะของนักศึกษาจะดับอุดมศึกษา กิจกรรมนอกหลักสูตรมีบทบาทสำคัญในการเสริมสร้างสมรรถนะที่จำเป็นทั้งในด้าน Soft Skills และ Hard Skills อย่างไรก็ตาม จากการศึกษาข้อมูลและสภาพแวดล้อมของการจัดกิจกรรมภายในมหาวิทยาลัย พบร่วมกันพบว่ามีปัญหาเชิงระบบและความท้าทายหลายประการที่ส่งผลต่อความต่อเนื่องและประสิทธิภาพในการติดตามพัฒนาการของนักศึกษา ดังนี้:

### ปัญหาที่ 1:

นักศึกษาไม่สามารถติดตามความก้าวหน้าของกิจกรรมที่ตนเข้าร่วมได้

กิจกรรมที่นักศึกษาเข้าร่วมมักเก็บข้อมูลแบบแยกส่วน และไม่มีระบบกลางสำหรับแสดงผลหรือวิเคราะห์ภาพรวมทำให้นักศึกษาไม่สามารถประเมินการพัฒนาของตนเองได้อย่างชัดเจน

**ความต้องการที่ตอบโจทย์:** ระบบที่สามารถรวมบันทึก และแสดงกิจกรรมที่นักศึกษาเข้าร่วมพร้อมรายละเอียดและทักษะที่ได้รับจากแต่ละกิจกรรม

### ปัญหาที่ 2:

นักศึกษาไม่ทราบว่าตนเองมีหรือขาดทักษะด้านใดบ้าง

ขาดระบบวิเคราะห์ที่สามารถเชื่อมโยงกิจกรรมกับชุดทักษะที่เกี่ยวข้อง เช่น ความสามารถด้านการสื่อสาร การเขียนโปรแกรม หรือการทำงานเป็นทีม ทำให้นักศึกษายากต่อการวางแผนพัฒนาตนเองในระยะยาว

**ความต้องการที่ตอบโจทย์:** ระบบวิเคราะห์ทักษะโดยอิงจากกิจกรรมที่เข้าร่วม และแสดงผลเปรียบเทียบกับกรอบทักษะตามหลักสูตรการศึกษา

### ปัญหาที่ 3:

อาจารย์ที่ปรึกษาไม่สามารถประเมินภาพรวมทักษะของนักศึกษาได้ง่าย

อาจารย์ไม่สามารถเข้าถึงข้อมูลกิจกรรมของนักศึกษาแบบเรียลไทม์ หรือในรูปแบบที่วิเคราะห์ได้ทันที ส่งผลให้การให้คำปรึกษาไม่ครอบคลุมหรือไม่ตรงจุด

**ความต้องการที่ตอบโจทย์:** ระบบ Dashboard สำหรับอาจารย์ที่ปรึกษา ซึ่งแสดงภาพรวมพัฒนาการทักษะรายบุคคลและรายชั้นปี เพื่อช่วยให้อาจารย์สามารถให้คำแนะนำแก่นักศึกษาได้อย่างตรงจุด

### ปัญหาที่ 4:

เจ้าหน้าที่กิจกรรมไม่สามารถวิเคราะห์แนวโน้มความนิยมและผลลัพธ์ของกิจกรรมได้อย่างมีประสิทธิภาพ ข้อมูลผู้สมัครและผู้เข้าร่วมกิจกรรมมักจัดเก็บกระจัดกระจาย ทำให้ยากต่อการวางแผนและปรับปรุงกิจกรรมในอนาคต

**ความต้องการที่ตอบโจทย์:** ระบบวิเคราะห์ข้อมูลกิจกรรมแบบรวมศูนย์ สามารถแสดงสถิติการเข้าร่วมกิจกรรมที่ความนิยม และประเมินคุณภาพกิจกรรมได้

ระบบ ANTivity จึงได้รับการออกแบบมาเพื่อแก้ไขปัญหาดังกล่าวแบบครบวงจร โดยมุ่งเน้นให้ผู้มีส่วนเกี่ยวข้องทั้งสามฝ่าย ได้แก่ นักศึกษา อาจารย์ที่ปรึกษา และเจ้าหน้าที่กิจกรรม สามารถใช้ระบบร่วมกันได้อย่างมีประสิทธิภาพ เพื่อส่งเสริมการพัฒนาทักษะของนักศึกษาอย่างเป็นรูปธรรมและต่อเนื่อง

## 1.5 ขอบเขตของโครงการ

โครงการระบบ ANTivity ถูกออกแบบและพัฒนาขึ้นเพื่อเป็นระบบสารสนเทศที่สนับสนุนการติดตามพัฒนาการทางทักษะของนักศึกษาจากการเข้าร่วมกิจกรรมนอกหลักสูตร โดยกำหนดขอบเขตของโครงการในระยะเริ่มต้นดังนี้:

### ขอบเขตด้านผู้ใช้งาน

ระบบรองรับการใช้งานจากผู้ใช้ 3 กลุ่มหลัก ได้แก่

**นักศึกษา (Student):** ลงทะเบียนเข้าสู่ระบบ ดูประวัติการเข้าร่วมกิจกรรม วิเคราะห์ทักษะที่ได้รับ สมัครเข้าร่วมกิจกรรม ประเมินกิจกรรม และยืนยันการเข้าร่วม

**อาจารย์ที่ปรึกษา (Professor):** เข้าดูข้อมูลภาพรวมพัฒนาการของนักศึกษารายบุคคลและรายชั้น ปี ตรวจสอบกิจกรรมที่เกี่ยวข้องกับนักศึกษา และใช้ข้อมูลสนับสนุนการให้คำปรึกษา

**เจ้าหน้าที่กิจกรรม (Staff):** จัดการกิจกรรม ทักษะ และหลักสูตรที่เกี่ยวข้อง อนุมัติผู้สมัครกิจกรรม ประเมินผู้เข้าร่วม และวิเคราะห์ข้อมูลเชิงสถิติ

### ขอบเขตด้านฟังก์ชันการทำงาน

- ระบบลงทะเบียนและเข้าสู่ระบบด้วย AWS Cognito เพื่อจัดการสิทธิ์และบทบาทผู้ใช้งาน
- การแสดงประวัติการเข้าร่วมกิจกรรมและทักษะที่ได้รับของนักศึกษา
- การเข้มโหยกิจกรรมกับชุดทักษะที่กำหนดไว้ในหลักสูตร
- การประเมินกิจกรรมโดยนักศึกษา และการประเมินผู้เข้าร่วมโดยเจ้าหน้าที่
- การแสดง Dashboard สรุปพัฒนาการทักษะของนักศึกษา และข้อมูลกิจกรรมสำหรับอาจารย์และเจ้าหน้าที่
- การเก็บและแสดงรูปภาพหรือเอกสารยืนยันผ่าน Amazon S3
- การจัดเก็บข้อมูลกิจกรรม ทักษะ ผู้ใช้งาน และผลการประเมินไว้ในฐานข้อมูล Amazon RDS

### ขอบเขตด้านเทคโนโลยี

- พัฒนา Frontend ด้วย Next.js
- พัฒนา Backend ด้วย Hono Framework ซึ่งทำงานบน Deno Runtime
- ใช้งานบริการ Cloud Infrastructure บน Amazon Web Services (AWS) เช่น EC2, S3, RDS, และ Cognito

### ขอบเขตด้านระยะเวลา

ระยะเวลาในการดำเนินโครงการครอบคลุมช่วงการวางแผน ออกแบบ พัฒนา ทดสอบ และจัดทำรายงานผล ภายในกรอบเวลาที่กำหนด

## บทที่ 2

### การออกแบบระบบและการใช้งาน

#### 2.1 บทบาทผู้ใช้งานและกรณีการใช้งาน (Use Case Scenarios)

ระบบ ANTivity ถูกออกแบบมาเพื่อรองรับการใช้งานของผู้ใช้ที่มีบทบาทแตกต่างกันจำนวน 3 กลุ่มหลัก ได้แก่ นักศึกษา (Student), อาจารย์ที่ปรึกษา (Professor) และ เจ้าหน้าที่ผู้จัดกิจกรรม (Staff) โดยแต่ละบทบาทมีหน้าที่และกรณีการใช้งานที่ตอบโจทย์ความต้องการเฉพาะด้าน ดังนี้:

##### 1. นักศึกษา (Student)

###### บทบาท:

ผู้ใช้งานระบบเพื่อสมัครเข้าร่วมกิจกรรม พัฒนาทักษะ และติดตามความก้าวหน้าของตนเอง

###### Use Case Scenarios:

- ลงทะเบียนบัญชีผู้ใช้ (Sign-up)
- เข้าสู่ระบบ (Login) และเข้าถึงหน้าໂປຣໄຟລ໌
- ตรวจสอบข้อมูลส่วนตัวและทักษะที่ได้รับจากกิจกรรม
- ค้นหาและเลือกกิจกรรมที่สนใจเข้าร่วม
- สมัครเข้าร่วมกิจกรรมผ่านระบบ
- ตรวจสอบรายการกิจกรรมที่เคยเข้าร่วม
- ยืนยันการเข้าร่วมในวันกิจกรรมจริง (หลังได้รับอนุมัติจากเจ้าหน้าที่)
- ประเมินกิจกรรมที่เข้าร่วม เพื่อให้ข้อเสนอแนะ

###### ประโยชน์ที่ได้รับ:

- เห็นพัฒนาการของตนเองในรูปแบบที่เป็นระบบ
- ทราบว่าตนมีหรือขาดทักษะใดบ้าง
- สามารถสมัครและติดตามกิจกรรมได้อย่างสะดวก
- ช่วยสะสมข้อมูลเพื่อประกอบ Portfolio ในอนาคต

##### 2. อาจารย์ที่ปรึกษา (Professor)

###### บทบาท:

ผู้ใช้งานระบบเพื่อติดตาม ตรวจสอบ และให้คำแนะนำในการพัฒนาทักษะของนักศึกษาในความดูแล

#### **Use Case Scenarios:**

- ลงทะเบียนและเข้าสู่ระบบในบทบาทอาจารย์
- เข้าถึง Dashboard ภาพรวมของนักศึกษาแต่ละคน
- วิเคราะห์พัฒนาการของนักศึกษารายชั้นปี
- เข้าดูໂປຣໄຟລ໌ເຊີງລຶກຂອງນักศึกษาแต่ละราย

#### **ประโยชน์ที่ได้รับ:**

- เข้าถึงข้อมูลนักศึกษาได้ตรงกับบทบาท
- มีข้อมูลสรุปแบบ Dashboard เพื่อการประเมินทักษะอย่างมีประสิทธิภาพ
- สามารถให้คำแนะนำแนะแนวตามข้อมูลพัฒนาการจริง
- มองเห็นภาพรวมทักษะของนักศึกษาในกลุ่มที่ดูแลอย่างชัดเจน

### **3. เจ้าหน้าที่ผู้จัดกิจกรรม (Staff)**

#### **บทบาท:**

ผู้ใช้งานระบบเพื่อบริหารจัดกิจกรรม ทักษะ หลักสูตร และจัดการข้อมูลผู้เข้าร่วม

#### **Use Case Scenarios:**

- ลงทะเบียนและเข้าสู่ระบบในบทบาทเจ้าหน้าที่
- จัดการรายการทักษะ (เพิ่ม/แก้ไข/ลบ)
- จัดการหลักสูตรและความสัมพันธ์กับทักษะ
- สร้าง แก้ไข ลบกิจกรรม
- ตรวจสอบและอนุมัติ/ปฏิเสธการสมัครเข้าร่วมกิจกรรม
- ยืนยันการเข้าร่วมจริง และให้ทักษะแก่ผู้เข้าร่วม
- วิเคราะห์ความนิยมของกิจกรรมในภาพรวม
- จัดการความสัมพันธ์ระหว่างนักศึกษาและอาจารย์ที่ปรึกษา

#### **ประโยชน์ที่ได้รับ:**

- บริหารกิจกรรมและหลักสูตรได้อย่างมีประสิทธิภาพ
- ติดตามข้อมูลแบบเรียลไทม์ ลดภาระการประสานงาน
- มีข้อมูลสนับสนุนการวางแผนกิจกรรมในอนาคต
- ป้องกันข้อผิดพลาดในการมอบหมายทักษะ
- เชื่อมโยงข้อมูลของนักศึกษาและอาจารย์ได้อย่างถูกต้อง

## 2.2 พัฟ์ชันหลักของระบบและประโยชน์ที่ได้รับ

ระบบ ANTivity ได้รับการออกแบบเพื่อให้บริการพัฟ์ชันหลักที่ครอบคลุมการทำงานของผู้ใช้งานกลุ่ม ได้แก่ นักศึกษา อาจารย์ที่ปรึกษา และเจ้าหน้าที่ผู้จัดกิจกรรม โดยมีพัฟ์ชันหลักที่สำคัญดังนี้:

### 1. พัฟ์ชันลงทะเบียนและยืนยันตัวตน (Authentication and Authorization)

- รองรับการสมัครสมาชิก (Sign-up) และเข้าสู่ระบบ (Login) ด้วยบัญชีแบบต่างกัน ได้แก่ Student, Professor และ Staff

- ใช้ AWS Cognito ใน การจัดการผู้ใช้ การยืนยันตัวตน และกำหนดสิทธิ์การเข้าถึงระบบตามบทบาท

#### ประโยชน์:

ช่วยให้ระบบมีความปลอดภัยสูง รองรับผู้ใช้หลายบทบาท และควบคุมการเข้าถึงข้อมูลอย่างเหมาะสม

### 2. พัฟ์ชันการจัดการกิจกรรม

- เจ้าหน้าที่สามารถสร้าง แก้ไข ลบ และเผยแพร่กิจกรรมที่ต้องการได้

- รองรับการดูรายละเอียดกิจกรรม ตรวจสอบผู้สมัคร อนุมัติ/ปฏิเสธการเข้าร่วม

- จัดเก็บรูปภาพหรือไฟล์ที่เกี่ยวข้องกับกิจกรรมใน Amazon S3

#### ประโยชน์:

ช่วยให้ฝ่ายกิจกรรมสามารถบริหารจัดการกิจกรรมได้สะดวกจากระบบเดียว ลดการใช้กระดาษ และตรวจสอบข้อมูลย้อนหลังได้

### 3. พัฟ์ชันการลงทะเบียนและติดตามกิจกรรมของนักศึกษา

- นักศึกษาสามารถค้นหา เลือก และสมัครเข้าร่วมกิจกรรมได้อย่างง่ายดาย

- แสดงรายการกิจกรรมที่เคยเข้าร่วม พร้อมรายละเอียดและทักษะที่ได้รับ

- ระบบรองรับการยืนยันการเข้าร่วมจริงและการประเมินกิจกรรมย้อนหลัง

#### ประโยชน์:

นักศึกษาสามารถวางแผนการเข้าร่วมกิจกรรมได้ตรงความสนใจ และติดตามพัฒนาการด้านทักษะของตนเองอย่างเป็นระบบ

### 4. พัฟ์ชันการวิเคราะห์และแสดงผลทักษะ

- ระบบเชื่อมโยงกิจกรรมกับทักษะที่เกี่ยวข้อง ทั้งในหมวด Soft Skills และ Hard Skills

- แสดงผลพัฒนาการด้านทักษะของนักศึกษาแบบ Dashboard

- แสดงสถานะว่า “มี”, “กำลังพัฒนา” หรือ “ยังขาด” ทักษะแต่ละด้าน

#### ประโยชน์:

ช่วยให้นักศึกษาและอาจารย์ที่ปรึกษาทราบสถานะพัฒนาการของนักศึกษาได้ชัดเจน และนำไปสู่การวางแผนพัฒนาต่อไปอย่างเหมาะสม

## 5. พึงก์ชันสรุปและวิเคราะห์ข้อมูลสำหรับอาจารย์ที่ปรึกษา

- แสดงสถิติพัฒนาการทักษะของนักศึกษารายบุคคลและรายชั้นปี
- สนับสนุนการให้คำปรึกษาเชิงลึกแบบรายบุคคล
- รองรับการถูรยละเอียดกิจกรรมที่นักศึกษาแต่ละคนเคยเข้าร่วม

ประโยชน์:

อาจารย์สามารถติดตามและประเมินนักศึกษาได้อย่างมีข้อมูลสนับสนุน และให้คำแนะนำเชิงกลยุทธ์ตามสมรรถนะรายบุคคล

## 6. พึงก์ชันการจัดการทักษะและหลักสูตร (สำหรับ Staff)

- เจ้าหน้าที่สามารถเพิ่ม แก้ไข ลบทักษะ และหลักสูตรที่เกี่ยวข้อง
- กำหนดความเข้มโดยง่ายระหว่างกิจกรรม ทักษะ และหลักสูตร
- ดูภาพรวมความก้าวหน้าของนักศึกษาในแต่ละหลักสูตร

ประโยชน์:

ช่วยให้ระบบมีความยืดหยุ่น รองรับนโยบายที่เปลี่ยนแปลง และสามารถออกแบบกิจกรรมที่ตอบสนองต่อเป้าหมายของหลักสูตร

## 7. พึงก์ชันวิเคราะห์แนวโน้มและประสิทธิภาพของกิจกรรม

- แสดงสถิติการเข้าร่วมกิจกรรม แยกตามประเภท ความนิยม และช่วงเวลา
- วิเคราะห์แนวโน้มการพัฒนาทักษะและระดับการมีส่วนร่วมของนักศึกษา

ประโยชน์:

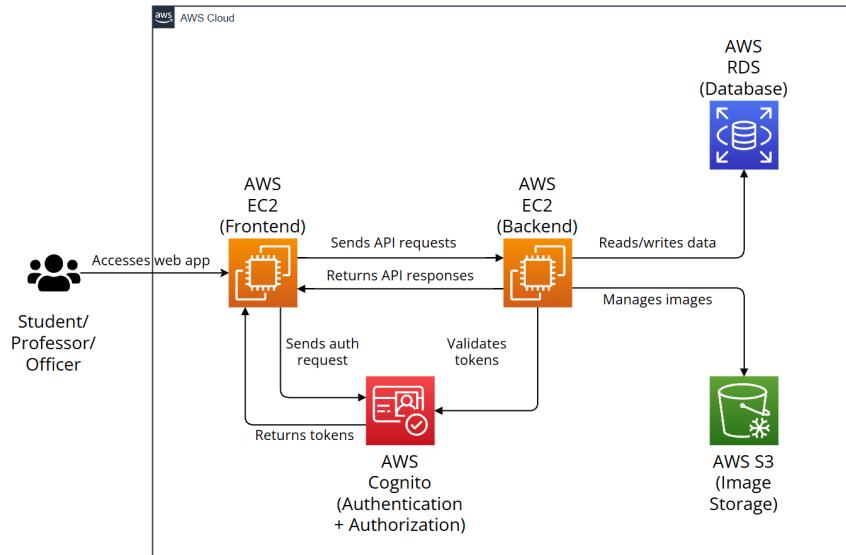
เจ้าหน้าที่สามารถใช้ข้อมูลวิเคราะห์ในการวางแผนกิจกรรมที่มีคุณภาพ สอดคล้องกับความต้องการของนักศึกษาและเป้าหมายการศึกษาในภาพรวม

ด้วยพึงก์ชันหลักข้างต้น ระบบ ANTivity จึงเป็นโซลูชันที่ตอบโจทย์ทั้งด้านการติดตามกิจกรรม การพัฒนาทักษะ และการจัดการข้อมูลเชิงวิเคราะห์อย่างครบวงจร โดยเน้นประสิทธิภาพ ความปลอดภัย และความสามารถในการปรับขยายในอนาคต

## บทที่ 3

### สถาปัตยกรรมระบบ

#### 3.1 ภาพสถาปัตยกรรมระบบโดยรวม



ภาพด้านบนแสดงโครงสร้างระบบ ANTivity ซึ่งเป็น Web Application ที่รองรับผู้ใช้งาน 3 บทบาทหลัก ได้แก่ นักศึกษา (Student), อาจารย์ที่ปรึกษา (Professor) และเจ้าหน้าที่กิจกรรม (Staff) โดยผู้ใช้จะเข้าถึงระบบผ่านเว็บเบราว์เซอร์ ซึ่งทำงานร่วมกับองค์ประกอบต่าง ๆ บน AWS Cloud ดังนี้:

- Amazon EC2 (Frontend) ให้บริการหน้าเว็บไซต์และส่วนติดต่อผู้ใช้งาน (User Interface)
- Amazon EC2 (Backend) ให้บริการ API และดำเนินการประมวลผลคำขอ
- AWS Cognito ใช้จัดการผู้ใช้และยืนยันตัวตน
- Amazon RDS จัดเก็บข้อมูลโครงสร้าง เช่น กิจกรรม ทักษะ ผู้ใช้
- Amazon S3 จัดเก็บไฟล์รูปภาพ เช่น รูปกิจกรรมและหลักฐานเข้าร่วม

ภาพแสดงการส่งผ่านของข้อมูลระหว่างองค์ประกอบต่าง ๆ ช่วยให้เห็นความสัมพันธ์เชิงเทคนิคและกระบวนการทำงานตั้งแต่ผู้ใช้งานเปิดเว็บไซต์ไปจนถึงการรับข้อมูลจากฐานข้อมูลหรือระบบจัดเก็บไฟล์

### 3.2 รายละเอียดบริการ AWS ที่ใช้

เพื่อให้ระบบสามารถดำเนินงานได้อย่างมีประสิทธิภาพ ความปลอดภัย และสอดคล้องกับแนวทางปฏิบัติของระบบสมัยใหม่ บริการหลักที่ใช้งาน AWS มีรายละเอียดดังนี้:

#### 1. Amazon EC2 (Frontend)

หน้าที่:

- ให้บริการหน้าเว็บของระบบ (Next.js) สำหรับผู้ใช้งานทุกบทบาท
- ทำหน้าที่รับคำขอจากผู้ใช้และแสดงข้อมูลที่ได้รับจาก Backend

ประโยชน์:

- ให้ประสบการณ์ผู้ใช้ที่รวดเร็ว
- แยกโหลดของ frontend และ backend อย่างอิสระ
- สามารถอัปเดต UI ได้โดยไม่กระทบกับระบบอื่น

#### 2. Amazon EC2 (Backend)

หน้าที่:

- พัฒนาโดยใช้ Hono Framework ซึ่งทำงานบน Deno Runtime
- ให้บริการ API แก่ Frontend สำหรับเรียกดูหรือเปลี่ยนแปลงข้อมูล
- ตรวจสอบ Token จาก Cognito ก่อนเข้าถึงฐานข้อมูลหรือจัดการไฟล์
- เชื่อมต่อกับ Amazon RDS (ผ่าน SQL) และ Amazon S3 (ผ่าน SDK)

ประโยชน์:

- ควบคุมสิทธิ์และความปลอดภัยของ API ได้ดี
- แยก concerns ระหว่าง data logic และ UI

#### 3. AWS Cognito

หน้าที่:

- บริหารจัดการการสมัครสมาชิก การเข้าสู่ระบบ และสิทธิ์ของผู้ใช้งาน
- สนับสนุนการแยกบทบาทผู้ใช้ (Role-based access control)
- ออก Token (JWT) ที่ใช้ในการตรวจสอบสิทธิ์ทุกครั้งที่ผู้ใช้เรียก API

ประโยชน์:

- เพิ่มความปลอดภัยในการควบคุมสิทธิ์
- ลดภาระการพัฒนา Auth เอง
- สามารถรองรับ Social Login (Google, Facebook) ได้ในอนาคต

#### 4. Amazon RDS (Relational Database Service)

หน้าที่:

ใช้ PostgreSQL สำหรับจัดเก็บข้อมูลโครงสร้าง เช่น:

- ข้อมูลผู้ใช้ (User Profile)
- รายการกิจกรรม (Activities)
- หักษะที่ได้รับ (Skill Mapping)
- ประวัติการเข้าร่วม (Participation)
- ผลการประเมินกิจกรรม (Feedback)

ประโยชน์:

- เก็บข้อมูลได้อย่างมีโครงสร้าง ช่วยให้ระบบสามารถทำการ Query, Filter และ Join ได้อย่างมีประสิทธิภาพ
- รองรับการเติบโตของข้อมูลในระยะยาว

#### 5. Amazon S3 (Simple Storage Service)

หน้าที่:

ใช้สำหรับจัดเก็บไฟล์รูปภาพ เช่น:

- รูปโปรไฟล์ผู้ใช้
- รูปกิจกรรม

ประโยชน์:

- รองรับไฟล์ขนาดใหญ่ได้ไม่จำกัด
- ปลอดภัย และสามารถกำหนดสิทธิ์การเข้าถึงรายไฟล์ได้ละเอียด
- ทำงานร่วมกับระบบ backend ได้อย่างยึดหยุ่น

### 3.3 การเชื่อมโยงและการทำงานร่วมกันของแต่ละองค์ประกอบ

กระบวนการสื้อสารและทำงานร่วมกันระหว่างองค์ประกอบของระบบเป็นไปในลักษณะดังนี้:

ลำดับการทำงานของระบบเมื่อผู้ใช้งานเข้าสู่ระบบ

1. การเข้าถึงเว็บไซต์

ผู้ใช้งานเปิดหน้าเว็บที่ให้บริการผ่าน Amazon EC2 (Frontend) ซึ่งโหลดส่วนติดต่อผู้ใช้ (UI)

2. การยืนยันตัวตน

ผู้ใช้งานกรอกข้อมูลตัวอิน → Frontend ส่งคำขอไปยัง AWS Cognito

Cognito ตรวจสอบข้อมูลและส่ง JWT Token กลับ

3. การเรียกใช้งานระบบหลังบ้านยืนยันตัวตน

Frontend ใช้ JWT Token ที่ได้แนบไปกับ API Request → ส่งไปยัง EC2 (Backend)

Backend ตรวจสอบ Token กับ Cognito และดำเนินการตามคำขอ เช่น:

อ่าน/เขียนข้อมูลจากฐานข้อมูล RDS

อัปโหลดหรืออัพเดตรูปภาพจาก S3

4. การตอบกลับผลลัพธ์

Backend ส่งข้อมูลหรือผลลัพธ์กลับมายัง Frontend เพื่อแสดงผลให้กับผู้ใช้งาน

แนวคิดในการแยกบทบาทและแยกความรับผิดชอบ

- ระบบแยก Frontend และ Backend ออกจากกัน ทำให้สามารถพัฒนาและขยายอิสระ

- ใช้ Token-based Authentication เทคนิคการเก็บ Session บนเซิร์ฟเวอร์

- ระบบฐานข้อมูล (RDS) และพื้นที่จัดเก็บไฟล์ (S3) แยกตามลักษณะข้อมูลเพื่อความยืดหยุ่น

### 3.4 ประโยชน์ของสถาปัตยกรรมนี้

**แยกส่วนชัดเจน (Separation of Concerns):** Frontend, Backend, Database, และ Storage ถูกแยกอิสระ เพื่อเพิ่มความยืดหยุ่นในการดูแลและขยายระบบ

**ปลอดภัย:** ใช้ Cognito สำหรับควบคุมสิทธิ์ และ JWT สำหรับตรวจสอบผู้ใช้ก่อนเข้าถึง API หรือข้อมูล

**ปรับขนาดได้ (Scalable):** EC2, RDS และ S3 สามารถปรับขยายตามปริมาณผู้ใช้งาน

**เหมาะสมกับระบบกิจกรรมที่เน้นการติดตามและวิเคราะห์:** เช่น การวิเคราะห์พัฒนาการทักษะ การแนะนำกิจกรรม และการจัดเก็บประวัติ

## บทที่ 4

### ขั้นตอนการติดตั้งและตั้งค่าระบบ

#### 4.1 การตั้งค่า AWS Cognito สำหรับการจัดการผู้ใช้

ระบบ ANTivity ใช้บริการ AWS Cognito เป็นเครื่องมือหลักในการจัดการการยืนยันตัวตน (Authentication) และการกำหนดสิทธิ์ (Authorization) สำหรับผู้ใช้งานแต่ละบุบทาท ได้แก่:

student – นักศึกษา

professor – อาจารย์ที่ปรึกษา

staff – เจ้าหน้าที่หรือผู้จัดกิจกรรม

การตั้งค่า AWS Cognito ประกอบด้วย 2 ส่วนหลัก ได้แก่ การสร้าง User Pool สำหรับบริหารจัดการผู้ใช้ และ การสร้าง App Clients เพื่อรองรับการเชื่อมต่อของระบบ Frontend และ Backend

##### ขั้นตอนที่ 1: สร้าง User Pool

1. เข้าสู่ AWS Management Console > บริการ Amazon Cognito

2. คลิก “Create user pool”

3. ตั้งค่าดังนี้:

Application type: Traditional web application

User pool name: ตั้งชื่อ เช่น antivity-user-pool

Sign-in options: เลือกใช้ Username และ Email

Required attributes: เลือก email

4. คลิก “Create user pool” เพื่อสร้าง

หมายเหตุ: หลังจากสร้างเสร็จจะได้รับค่า User Pool ID และ Region ซึ่งจะต้องนำไปใช้ในไฟล์ .env

##### ขั้นตอนที่ 2: สร้าง User Groups สำหรับแยกบทบาทผู้ใช้

1. ภายใน User Pool ที่สร้างไว้ ให้ไปที่แท็บ “Groups”

2. กดปุ่ม “Create group” และสร้างกลุ่มผู้ใช้ตามนี้:

student

professor

staff

ตั้งชื่อกลุ่มด้วยตัวพิมพ์เล็กทั้งหมด เพื่อให้สอดคล้องกับระบบที่กำหนดไว้ล่วงหน้า โดยผู้ใช้งานแต่ละคนจะถูกจัดให้อยู่ในกลุ่มตามบทบาทของตนเมื่อสมัครสมาชิกหรือผ่านการอนุมัติ

### ขั้นตอนที่ 3: สร้าง App Clients

#### สำหรับ Backend

1. ไปที่แท็บ App clients > คลิก Create app client
2. ตั้งชื่อ เช่น antivity-backend-client
3. Application type: Traditional web application
4. เปิดใช้งาน Generate client secret
5. คลิก Create → ระบบจะให้ Client ID และ Client secret

#### สำหรับ Frontend

1. สร้างอีก Client โดยตั้งชื่อ เช่น antivity-frontend-client
2. Application type: Single-page application (SPA)
3. ไม่เปิดใช้ Client secret (เนื่องจากใช้บันผ่าน client)
4. คลิก Create → ได้ Client ID

### ขั้นตอนที่ 4: เขียนไฟล์ .env

ไฟล์ .env ของ Backend

```
COGNITO_USER_POOL_ID=us-east-1_XXXXXXX  
COGNITO_CLIENT_ID=xxxxxxxxxxxxxxxxxxxxxx  
COGNITO_CLIENT_SECRET=xxxxxxxxxxxxxxxxxxxxxx  
COGNITO_REGION=us-east-1
```

ไฟล์ .env ของ Frontend

```
NEXT_PUBLIC_COGNITO_USER_POOL_ID=us-east-1_XXXXXXX  
NEXT_PUBLIC_COGNITO_USER_POOL_CLIENT_ID=xxxxxxxxxxxxxx
```

#### การทำงานของระบบร่วมกับ Cognito

1. ผู้ใช้งานทะเบียนผ่านหน้าเว็บ → Frontend ส่งข้อมูลไปยัง AWS Cognito
2. หลังสมัครเสร็จ ระบบสามารถจัดผู้ใช้เข้าสู่กลุ่ม (student, professor, หรือ staff) ได้โดยอัตโนมัติ
3. ผู้ใช้เข้าสู่ระบบและรับ JWT Token จาก Cognito
4. Token ดังกล่าวจะถูกแนบไปกับทุก API Request → Backend ตรวจสอบกับ Cognito
5. Backend ใช้ข้อมูลจาก Token เช่น cognito:groups เพื่อจำแนกสิทธิ์และบทบาทของผู้ใช้ในระบบ

## ประโยชน์จากการใช้ Cognito + Group

- แยกสิทธิ์ใช้ตามกลุ่ม (student, professor, staff) อย่างปลอดภัยและชัดเจน
- รองรับการขยายบทบาทเพิ่มเติมในอนาคต เช่น admin, external
- ไม่จำเป็นต้องพัฒนา authentication system เอง ลดภาระการดูแลข้อมูลสำคัญ

## 4.2 การตั้งค่า Amazon RDS และการออกแบบฐานข้อมูล

ระบบ ANTivity ใช้บริการ Amazon RDS สำหรับจัดการฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database) โดยใช้ PostgreSQL เป็นระบบฐานข้อมูลหลัก เนื่องจากรองรับความสามารถในการใช้งานร่วมกับ UUID, Constraints และ TimeStamp ซึ่งเหมาะสมสำหรับงานเชิงระบบในระดับองค์กร

การตั้งค่า RDS และโครงสร้างฐานข้อมูลของระบบมีรายละเอียดดังนี้:

### ขั้นตอนที่ 1: การสร้างฐานข้อมูล RDS (PostgreSQL)

1. เข้าสู่ AWS Management Console > Amazon RDS > Create database
2. เลือกค่าต่อไปนี้:

Database creation method: Standard create

Engine type: PostgreSQL

Templates: Free tier

DB instance identifier: ตั้งชื่อ เช่น antivity-db

Master username / password: ตั้งค่าการเข้าสู่ระบบฐานข้อมูล

Public access: Yes

VPC security group: เลือกหรือสร้างกลุ่ม default ที่อนุญาตพอร์ต 5432

3. เพิ่ม Inbound Rule ให้กับ Security Group:

Type: PostgreSQL

Protocol: TCP

Port range: 5432

Source: 0.0.0.0/0 (สำหรับการทดสอบภายใน ควรปรับให้ปลอดภัยในการใช้งานจริง)

## ขั้นตอนที่ 2: การตั้งค่าไฟล์ .env สำหรับเชื่อมต่อ RDS

ภายในโปรเจกต์ Backend ให้สร้างไฟล์ .env และระบุค่าการเชื่อมต่อฐานข้อมูลดังนี้:

```
PGHOST=<Endpoint ของ RDS จาก AWS Console>
```

```
PGPORT=5432
```

```
PGUSERNAME=<Master username>
```

```
PGPASSWORD=<Master password>
```

```
PGDATABASE=postgres
```

## ขั้นตอนที่ 3: การออกแบบฐานข้อมูลและการสร้างตาราง

ระบบ ANTivity ไม่ใช้การสร้างตารางแบบแม่นวนฝ่าน SQL Console แต่ใช้โค้ดที่ผัง Backend เป็นผู้สร้าง อัตโนมัติเมื่อระบบทำงาน โดยใช้ CREATE TABLE IF NOT EXISTS สำหรับสร้างตารางทั้งหมดจำนวน 12 ตารางหลัก ดังนี้:

ตารางผู้ใช้และบทบาท

```
student
```

```
professor
```

```
professor_student (ความสัมพันธ์ระหว่างอาจารย์-นักศึกษา)
```

ตารางกิจกรรมและการเข้าร่วม

```
activity
```

```
student_activity
```

```
activity_evaluation (การประเมินกิจกรรม)
```

ตารางทักษะและหลักสูตร

```
skill
```

```
student_skill
```

```
student_skill_log
```

```
curriculum
```

```
curriculum_skill
```

```
activity_skill
```

## ความปลอดภัยและความถอดรหัส

- ใช้ Constraints และ Foreign Key ครบถ้วน ป้องกันข้อมูลซ้ำ/ไม่สัมพันธ์

- ใช้ CHECK ในหลายตาราง เช่น status, gender, level เพื่อกำหนดค่าไว้ถูกต้องหรือไม่

- ใช้ ON DELETE CASCADE ในความสัมพันธ์หลัก เพื่อให้ระบบจัดการข้อมูลอย่างถูกต้องเมื่อลบข้อมูลแม่

#### 4.3 การตั้งค่า Amazon S3 สำหรับจัดเก็บไฟล์

ระบบ ANTivity ใช้บริการ Amazon Simple Storage Service (Amazon S3) สำหรับจัดเก็บไฟล์ประเภทต่าง ๆ ที่เกี่ยวข้องกับกิจกรรมและผู้ใช้งาน เช่น:

รูปภาพกิจกรรม

รูปโปรไฟล์นักศึกษาและอาจารย์

Amazon S3 เป็นบริการ object storage ที่มีความเสถียร ปลอดภัย และสามารถรองรับการเข้าถึงข้อมูลผ่าน API ได้อย่างรวดเร็ว เหนาะสำหรับใช้ร่วมกับ Web Application ที่ต้องการความสามารถในการจัดเก็บไฟล์บนคลาวด์

##### ขั้นตอนที่ 1: การสร้าง S3 Bucket

1. เข้าสู่ AWS Management Console > บริการ Amazon S3

2. คลิก “Create bucket”

3. ตั้งค่าดังนี้:

Bucket name: ตั้งชื่อ เช่น antivity-assets (ต้องไม่ซ้ำกับ Bucket อื่นใน AWS Global)

Region: เลือกให้ตรงกับ Region ที่ใช้ในบริการอื่น (เช่น us-east-1)

Block all public access: ยกเลิกการติ๊กตัวเลือกนี้ แล้วติ๊ก "I acknowledge..." เพื่่อนญาตการเข้าถึงแบบสาธารณะสำหรับไฟล์ที่ต้องการ

4. กด “Create bucket”

##### ขั้นตอนที่ 2: การตั้งค่า Bucket Policy (อนุญาตให้โหลดไฟล์แบบสาธารณะ)

เพื่อให้ระบบสามารถโหลดรูปภาพผ่าน URL โดยไม่ต้องยืนยันตัวตน ให้ตั้งค่า Bucket Policy ดังนี้:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadGetObject",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::antivity-assets/*"  
        }  
    ]  
}
```

\*ให้แทน antivity-assets ด้วยชื่อ Bucket ที่ตั้งไว้ของจริง

### ขั้นตอนที่ 3: การตั้งค่า CORS (Cross-Origin Resource Sharing)

เพื่อให้เว็บแอปพลิเคชันสามารถเรียกดูและอัปโหลดไฟล์ข้ามโดเมนได้ ให้ตั้งค่า CORS ของ Bucket ดังนี้:

```
[  
 {  
     "AllowedHeaders": ["*"],  
     "AllowedMethods": ["GET", "PUT", "POST", "DELETE"],  
     "AllowedOrigins": ["*"],  
     "ExposeHeaders": [],  
     "MaxAgeSeconds": 3000  
 }  
 ]
```

สามารถปรับ AllowedOrigins ให้เฉพาะจุดเมนจริงของระบบในภายหลังเพื่อความปลอดภัย

### ขั้นตอนที่ 4: การเชื่อมต่อ S3 ในระบบ Backend

```
AWS_ACCESS_KEY_ID=<Access Key ID จาก IAM หรือ temporary credential>  
AWS_SECRET_ACCESS_KEY=<Secret Access Key>  
AWS_SESSION_TOKEN=<หากใช้แบบ temporary credential>  
AWS_REGION=us-east-1  
AWS_S3_BUCKET=antivity-assets
```

ค่าดังกล่าวจะถูกนำไปใช้กับ AWS SDK ในฝั่ง Backend เพื่ออัปโหลดและดึงไฟล์จาก S3

ตัวอย่างการใช้งานในระบบ

- เมื่อเจ้าหน้าที่เพิ่มกิจกรรมใหม่ ระบบสามารถอัปโหลดรูปหน้าปกกิจกรรมไปยัง S3 และบันทึก URL กลับไปยัง

#### ฐานข้อมูล

- เมื่อนักศึกษาเข้าดูรายละเอียดกิจกรรม ระบบจะโหลดรูปภาพจาก URL ที่จัดเก็บไว้ ซึ่งซึ่งไปยังไฟล์ใน S3
- รูปภาพสามารถเข้าถึงผ่าน <https://<bucket-name>.s3.amazonaws.com/<filename>> ได้ทันที หากถูกกำหนดให้เป็น public

## 4.4 การติดตั้งและดีเพลโลยแอป (Next.js + Hono บน EC2)

เพื่อให้ระบบสามารถให้บริการผ่านเว็บไซต์ได้จริง ผู้พัฒนาได้ทำการติดตั้งและดีเพลโลยส่วนประกอบทั้งสองของระบบ คือ Frontend (Next.js) และ Backend (Hono + Deno) บนเซิร์ฟเวอร์ Amazon EC2 โดยแยกอินสแตนซ์กันอย่างชัดเจน ทั้งนี้เพื่อลดความซับซ้อนในการดูแลรักษา และสามารถปรับขยายในอนาคตได้อย่างมีประสิทธิภาพ รายละเอียดขั้นตอนการติดตั้งและดีเพลโลยแอปมีดังนี้:

### 1. การดีเพลโลยระบบ Backend (Hono + Deno บน EC2)

#### 1.1 สร้างอินสแตนซ์ EC2

เลือก Amazon Linux 2 เป็น AMI

ประเภทอินสแตนซ์: t2.micro (Free tier)

สร้าง Key Pair สำหรับการเชื่อมต่อ SSH

ตั้งค่า Security Group ให้เปิดพอร์ต 80 และ 443

#### 1.2 เชื่อมต่อและติดตั้ง Deno

```
ssh -i "ชื่อไฟล์-key.pem" ec2-user@<Public-IP>
```

```
curl -fsSL https://deno.land/install.sh | sh
```

```
echo 'export PATH="$HOME/.deno/bin:$PATH"' >> ~/.bashrc
```

```
source ~/.bashrc
```

#### 1.3 อัปโหลดและติดตั้งระบบ

```
scp -i "ชื่อไฟล์-key.pem" backend.zip ec2-user@<Public-IP>:~
```

```
unzip backend.zip
```

```
cd backend/
```

#### 1.4 สร้างไฟล์ .env และกำหนดตัวแปร

ประกอบด้วยข้อมูลเชื่อมต่อ RDS, Cognito และ S3 ดังที่ระบุไว้ในหัวข้อก่อนหน้า

#### 1.5 เรียกใช้งานระบบ

```
deno run --watch --allow-net --env-file=.env --allow-env --allow-write --allow-read --allow-sys main.ts
```

## 2. การดีเพลย์ระบบ Frontend (Next.js บน EC2)

### 2.1 สร้างอินสแตนซ์ EC2

ใช้ Amazon Linux 2 และ t2.micro

เปิดพอร์ต 80 และ 443

สร้าง Key Pair สำหรับการ SSH

### 2.2 เชื่อมต่อและติดตั้ง Node.js

```
ssh -i "ชื่อไฟล์-key.pem" ec2-user@<Public-IP>
```

```
curl -fsSL https://rpm.nodesource.com/setup_18.x | sudo bash -
```

```
sudo yum install -y nodejs
```

### 2.3 อัปโหลดและติดตั้งระบบ

```
scp -i "ชื่อไฟล์-key.pem" frontend.zip ec2-user@<Public-IP>:~
```

```
unzip frontend.zip
```

```
cd frontend/
```

```
npm install
```

### 2.4 สร้างไฟล์ .env สำหรับ Frontend

เช่น:

```
NEXT_PUBLIC_API_BASE_URL=http://<Public-IP-Address-of-Backend>
```

```
NEXT_PUBLIC_COGNITO_USER_POOL_ID=<UserPoolId>
```

```
NEXT_PUBLIC_COGNITO_USER_POOL_CLIENT_ID=<FrontendAppClientId>
```

### 2.5 เรียกใช้งาน

```
sudo npx next start -H 0.0.0.0 -p 80
```

คำสั่งนี้จะรัน Next.js บนพอร์ต 80 และเปิดให้เข้าถึงจาก IP ภายนอก

### การเชื่อมโยงระบบทั้งหมด

- Frontend ทำหน้าที่ติดต่อกับผู้ใช้งานผ่าน Browser และส่งคำขอ API ไปยัง Backend

- Frontend เรียก AWS Cognito เพื่อรับ Token

- Backend ตรวจสอบ Token กับ Cognito ก่อนดำเนินการใด ๆ

- Backend ดึง/บันทึกข้อมูลจาก Amazon RDS และ Amazon S3 ตามความจำเป็น

ทั้ง EC2 Backend และ Frontend สามารถเชื่อมต่อด้วย Load Balancer หรือกำหนด Auto Scaling Group ได้ในอนาคต

## บทที่ 5

### ผลการทดสอบระบบและการรับมือกับข้อผิดพลาด

#### 5.1 รายงานผลการทดสอบแบบ End-to-End

การทดสอบระบบ ANTivity ได้ดำเนินการใน 3 สถานการณ์ที่สอดคล้องกับลักษณะการใช้งานจริง โดยครอบคลุมผู้ใช้งาน 3 บทบาทหลัก ได้แก่ นักศึกษา อาจารย์ที่ปรึกษา และเจ้าหน้าที่กิจกรรม เพื่อยืนยันว่าระบบสามารถตอบสนองต่อการใช้งานได้อย่างสมบูรณ์ และมีการจัดการกับข้อผิดพลาดของผู้ใช้งานมีประสิทธิภาพ

##### กรณีที่ 1: นักศึกษาเข้าสู่ระบบและตรวจสอบทักษะของตนเอง

###### เป้าหมายการทดสอบ:

ตรวจสอบว่านักศึกษาสามารถเข้าสู่ระบบได้อย่างถูกต้อง และสามารถดูพัฒนาการด้านทักษะของตนเองที่ได้รับจากกิจกรรมต่าง ๆ ได้ครบถ้วน

###### ขั้นตอนการทดสอบ:

1. เข้าสู่ระบบด้วยบัญชี Cognito ที่ลงทะเบียนไว้ล่วงหน้า
2. ระบบนำเข้าสู่หน้า "โปรไฟล์ของฉัน" และโหลดข้อมูลกิจกรรมและทักษะ พร้อมແທບເປົ້ອເຊັນຄວາມคືບໜ້າ

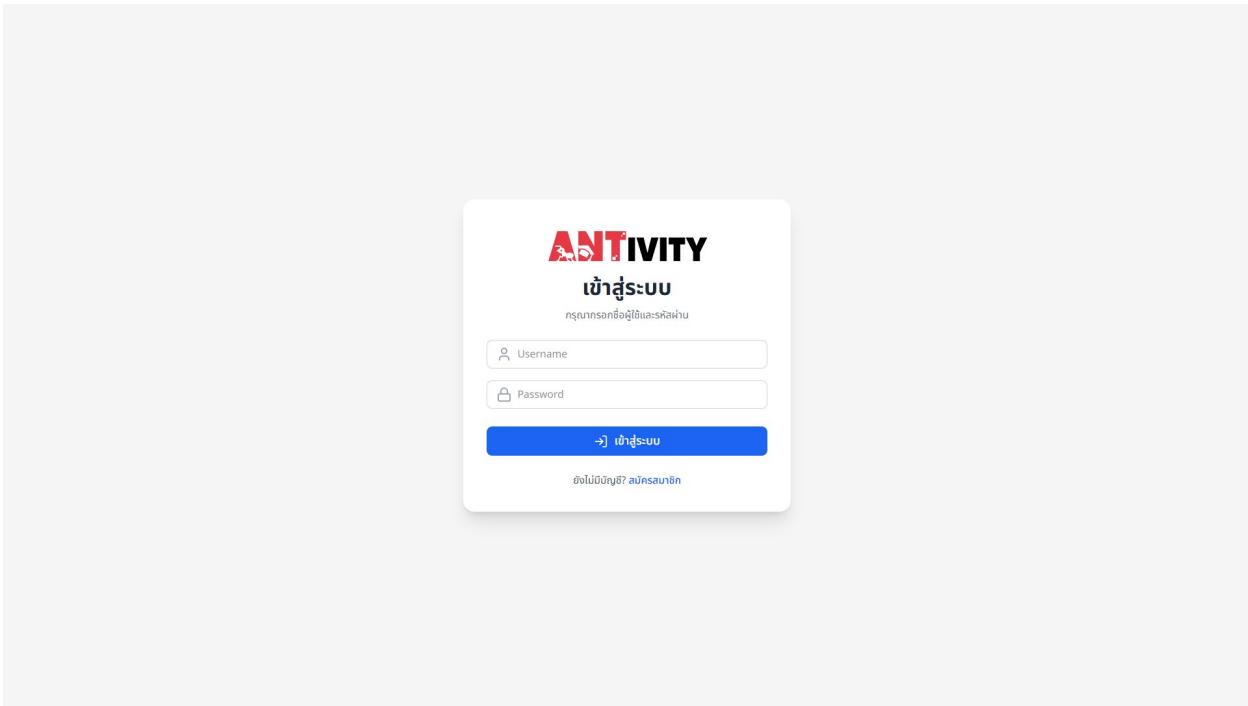
3. ระบบแสดงรายการกิจกรรมที่เคยเข้าร่วม พร้อมทักษะที่ได้จากแต่ละกิจกรรม
4. เข้าหน้ากิจกรรมของฉันเพื่อดูและตรวจสอบประวัติการเข้าร่วมกิจกรรม

###### ผลลัพธ์:

- ระบบโหลดข้อมูลได้เร็วและไม่มีข้อผิดพลาด
- ระบบแสดงผลได้รวดเร็ว และไม่มีข้อผิดพลาดในการโหลด
- ข้อมูลกิจกรรมตรงกับฐานข้อมูลจริง
- UI มีการแยกหมวดหมู่ข้อมูลอย่างชัดเจน
- ผู้ใช้สามารถประเมินพัฒนาการของตนเองได้ง่าย

###### รูปภาพประกอบ:





รูปที่ 5.1 หน้า login

รูปที่ 5.2 หน้าprofileนักศึกษา พร้อมกิจกรรมทักษะ

**กิจกรรมของฉัน**

( กิจกรรม (1) | ตรวจสอบ (0) | ยังไม่รับข้อสอบ (0) | ยังไม่ตั้งเป้า (0) | ไม่เข้าร่วม (0) | ไม่อนุญาต (0) | เข้าร่วมแล้ว (1) | ไม่ได้เข้าร่วม (0) )

**อบรม Java เมื่อชั้น**

28 พฤษภาคม 2568  
สอนพื้นฐาน Java  
 สามารถอธิบาย: เครื่องเรียน  
 สามารถอธิบาย: เผื่อนแผลลีว  
 สามารถอธิบาย: ขั้นตอนแล้ว  
 สามารถประเมินกิจกรรม: ประเมินแล้ว  
 แบบประเมินกิจกรรม: สังเคราะห์

**ดูรายละเอียด**

รูปที่ 5.3 หน้ากิจกรรมของฉัน

**กิจกรรม “อบรมการทำงานเป็นทีม”**

หลักสูตร “อบรมการทำงานเป็นทีม” TEAMWORK

**ข้อมูลกิจกรรม**

รายละเอียด: เปิดวิทยาลัยเทคโนโลยี  
 สถานะ: เริ่มแล้ว  
 จำนวนผู้เข้าร่วม: 1/10  
 ค่าใช้: us2  
 สร้างเมื่อ: 23 พฤษภาคม 2568  
 อัปเดตล่าสุด: 23 พฤษภาคม 2568

**กำหนดการ**

งวดที่ 1 กิจกรรม: 28 พฤษภาคม 2568  
 ใบสมัครเข้าร่วม: 25 พฤษภาคม 2568  
 ยังไม่ตั้งเป้า: 23 พฤษภาคม 2568 ถึง 27 พฤษภาคม 2568

**กิจกรรมที่จะได้รับ**

การทักษะเป็นทีม (soft) - ระดับ 2 | การมีความรู้ Java (hard) - ระดับ 4

กลับ | เผื่อนแผลลีว |

รูปที่ 5.4 หน้าแสดงรายละเอียดกิจกรรมที่เคยเข้าร่วม

## กรณีที่ 2: อาจารย์ที่ปรึกษาดูพัฒนาการของนักศึกษาที่ดูแล

### เป้าหมายการทดสอบ:

ตรวจสอบว่าอาจารย์สามารถเข้าถึงข้อมูลนักศึกษาที่ตนรับผิดชอบได้อย่างถูกต้อง และใช้ข้อมูลนั้นในการประเมินพัฒนาการได้จริง

### ขั้นตอนการทดสอบ:

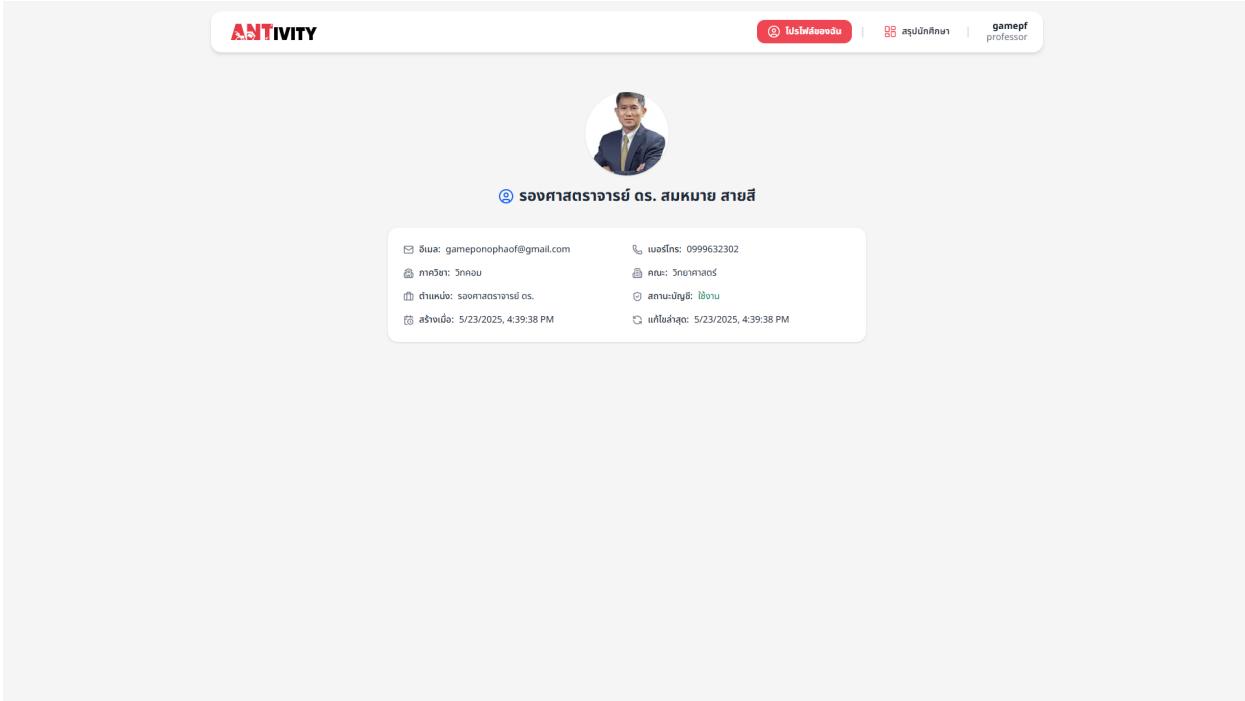
1. อาจารย์ล็อกอิน
2. เข้าสู่หน้า “สรุปนักศึกษา” เพื่อตรวจสอบรายชื่อนักศึกษาในความดูแล
3. เลือกดูรายงานเชิงลึกของนักศึกษารายบุคคล
4. ตรวจสอบข้อมูลกิจกรรมและทักษะที่ได้รับจากระบบ

### ผลลัพธ์:

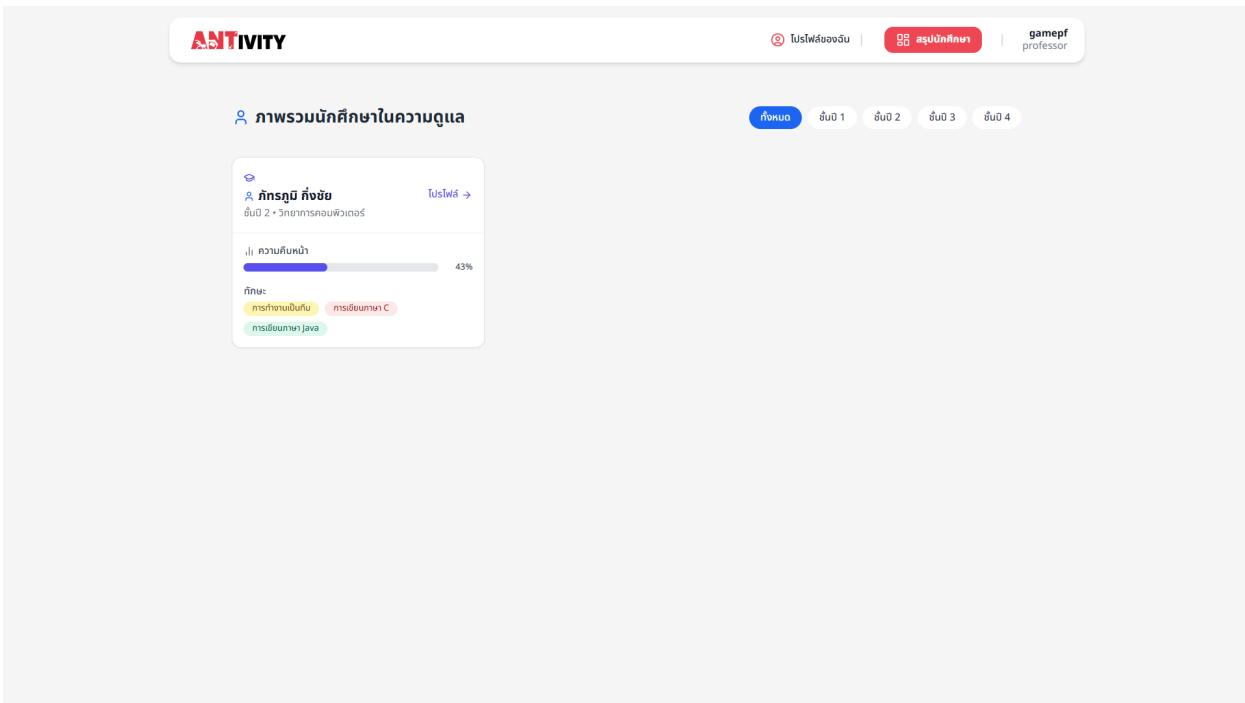
- ระบบแสดงข้อมูลแบบ Dashboard ชัดเจน
- มีการแสดงเปอร์เซ็นต์ความสำเร็จของทักษะเทียบกับหลักสูตร
- รองรับการดูข้อมูลรายบุคคลอย่างละเอียด
- ไม่มีการโหลดข้อมูลผิดหรือแสดงข้อมูลซ้ำ

### รูปภาพประกอบ:





ຮູບທີ 5.5 ພັນຍາແສດງຮາຍລະເວີຍດົກຈິກຮຽນທີ່ເຄີຍເຂົ້າຮ່ວມ



ຮູບທີ 5.6 ສຽງນັກສຶກສາໃນທີ່ປັບປຸງ

The screenshot shows the ANTIIVITY platform interface. At the top, there are navigation links: 'ไปรษณีย์' (Post Office), 'ไปรษณีย์อิเล็กทรอนิกส์' (Electronic Postage), 'ลงทะเบียนกิจกรรม' (Register Activity), 'กิจกรรมของฉัน' (My Activities), and 'student'. Below this, a user profile for 'ภัทรฤมิ กั่งขัย' (Phattaraphum) is displayed, showing contact information like email (phattaraphum.cth@gmail.com), phone number (0999632302), LINE ID (gamephat), and gender (male). A progress bar indicates 43% completion (3/7 หน่วย). To the right, there are sections for 'กิจกรรมที่กำลังเรียน' (Activities being learned), 'กิจกรรมที่สำเร็จ' (Completed activities), and 'กิจกรรมที่ไม่สำเร็จ' (Incomplete activities). Below the profile, a news feed item is shown: 'กิจกรรมที่เคยเข้าร่วม' (Activities participated in) featuring a thumbnail for 'อบรม Java เมื่อวัน' (Java Training on [date]) with 28 participants.

รูปที่ 5.7 รายละเอียดนักศึกษารายบุคคล

กรณีที่ 3: เจ้าหน้าที่สร้างกิจกรรมแต่ลืมกรอกทักษะที่เกี่ยวข้อง (กรณีแสดงข้อผิดพลาดและการรับมือ)

#### เป้าหมายการทดสอบ:

ตรวจสอบว่าเมื่อเจ้าหน้าที่กรอกข้อมูลกิจกรรมไม่ครบถ้วน (เช่น ไม่ระบุทักษะ) ระบบสามารถตรวจจับข้อผิดพลาดและแสดงคำเตือนได้ก่อนบันทึก

#### ขั้นตอนการทดสอบ:

- เจ้าหน้าที่เข้าสู่ระบบและไปที่หน้า “จัดการกิจกรรม”
- กรอกรายละเอียดกิจกรรมแต่ ลืมกรอกส่วนกำหนดการ & สถานที่
- กด “บันทึกกิจกรรม”

#### สิ่งที่เกิดขึ้น:

ระบบแสดงหน้าต่างแจ้งเตือนทันที:

“กรุณากรอกข้อมูลให้ครบถ้วน”

พร้อมรายการช่องที่จำเป็นต้องกรอก และขอบเขตช่องที่จำเป็นต้องกรอก

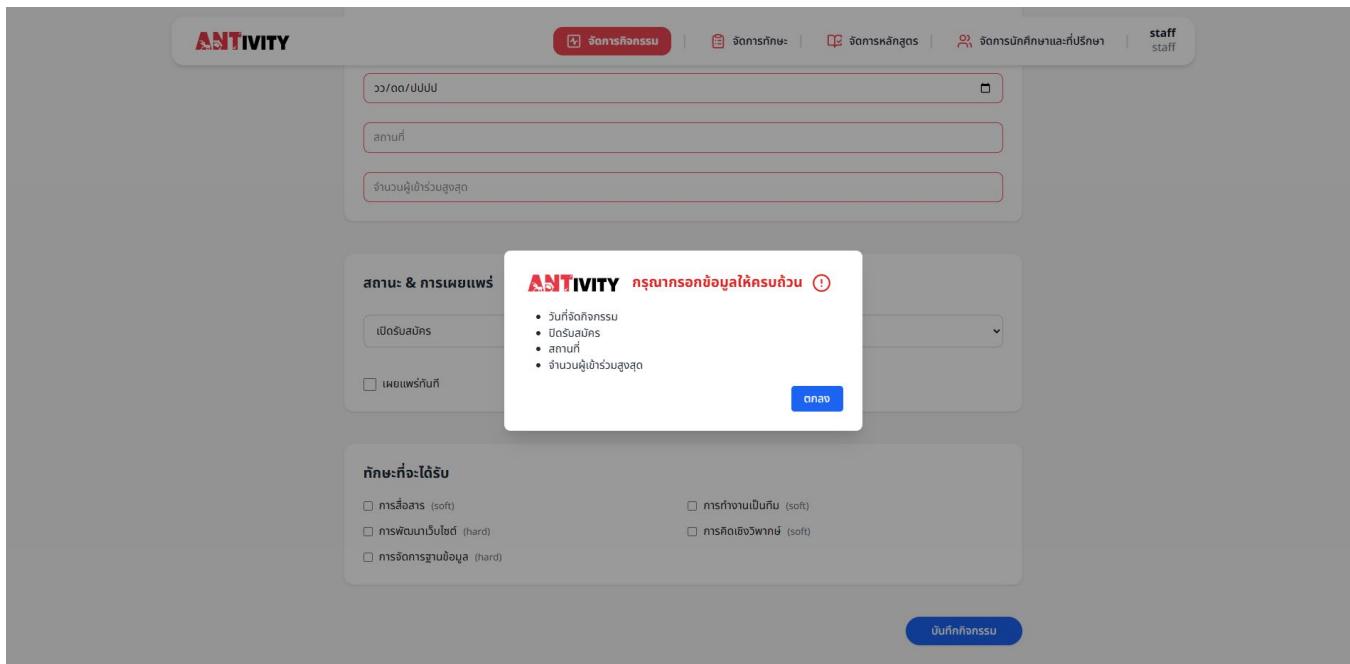
#### ผลลัพธ์:

- ระบบสามารถตรวจจับข้อผิดพลาดผ่านฟังผู้ใช้ (Validation) ได้อย่างถูกต้อง
- ป้องกันข้อมูลไม่ครบถูกบันทึกเข้าสู่ระบบ
- ไม่มีข้อมูลผิดพลาดถูกส่งไปยัง Backend

## รูปภาพประกอบ:

รูปที่ 5.8 การแสดงหน้าต่าง “กรุณากรอกข้อมูลให้ครบถ้วน”

รูปที่ 5.9 ขอบແດງທີ່ໜ່ອງແສດງຄືງຄວາມຈຳເປັນຕ້ອງກະລຸນາກ



รูปที่ 5.8 การแสดงหน้าต่าง “กรุณากรอกข้อมูลให้ครบถ้วน”



รูปที่ 5.9 ขอบແດງທີ່ໜ່ອງແສດງຄືງຄວາມຈຳເປັນຕ້ອງກະລຸນາກ

## 5.2 กรณีที่ระบบทำงานปกติ

จากการทดสอบระบบในหลายกรณีโดยครอบคลุมผู้ใช้งานทั้งสามบทบาท ได้แก่ นักศึกษา อาจารย์ที่ปรึกษา และเจ้าหน้าที่กิจกรรม พบร่วมกันว่าระบบสามารถทำงานได้อย่างต่อเนื่องและน่าเชื่อถือ ภายใต้เงื่อนไขที่ผู้ใช้งานกรอกข้อมูลอย่างครบถ้วนและถูกต้อง

ตัวอย่างผลการทำงานปกติ:

นักศึกษาสามารถเข้าสู่ระบบและดูข้อมูลกิจกรรมที่เข้าร่วมได้อย่างถูกต้อง พร้อมแสดงทักษะที่ได้รับ และระดับความสำเร็จของแต่ละทักษะในรูปแบบ Dashboard ที่เข้าใจง่าย

อาจารย์ที่ปรึกษาสามารถติดตามพัฒนาการของนักศึกษาในความดูแลได้แบบเรียลไทม์ โดยมีการแสดงความคืบหน้าทักษะเบรียบเทียบกับเกณฑ์ของหลักสูตร

เจ้าหน้าที่สามารถเพิ่ม แก้ไข และจัดการกิจกรรมได้อย่างราบรื่น รวมถึงสามารถจัดสรรทักษะที่เหมาะสมให้กับกิจกรรมแต่ละประเภทได้ครบถ้วน

การเชื่อมต่อระหว่างระบบ Frontend และ Backend ผ่าน API มีความเสถียร ไม่พบอาการค้าง (hang) หรือโหลดข้อมูลล่าช้า

ความน่าเชื่อถือ (Reliability):

- การดึงข้อมูลผู้ใช้งานผ่าน Cognito เป็นไปอย่างปลอดภัยและแม่นยำ
- การสื่อสารกับฐานข้อมูลผ่าน RDS และการจัดเก็บไฟล์บน S3 ดำเนินไปอย่างไร้ข้อผิดพลาด
- ไม่มีเหตุการณ์ downtime หรือ error 500 ภายใต้การใช้งานปกติ

## 5.3 กรณีที่เกิดข้อผิดพลาดและระบบรับมือ

ในกรณีที่ผู้ใช้งานกรอกข้อมูลไม่ครบหรือกรอกไม่ถูกต้อง ระบบมีการออกแบบให้สามารถป้องกันข้อผิดพลาดได้อย่างมีประสิทธิภาพ โดยใช้เทคนิค Client-side Validation และ Backend validation ควบคู่กัน เพื่อป้องกันข้อมูลที่ไม่สมบูรณ์เข้าสู่ระบบฐานข้อมูล

ตัวอย่างข้อผิดพลาดและการจัดการ:

**กรณี 1: ลืมกรอก “ข้อมูลกิจกรรมไม่ครบถ้วน” ขณะสร้างกิจกรรม**

สถานการณ์: เจ้าหน้าที่กรอกข้อมูลกิจกรรมใหม่แต่ไม่ได้กรอกข้อมูลให้ครบถ้วน

การตอบสนองของระบบ:

ระบบแสดงข้อความเตือนในรูปแบบ banner:

“กรุณากรอกข้อมูลให้ครบถ้วน”

พร้อมรายการซ่อนที่จำเป็นต้องกรอก และขอบเขตของที่จำเป็นต้องกรอก

## กรณี 2: พยายามเข้าสู่ระบบด้วยบัญชีที่ไม่มีใน Cognito

การตอบสนองของระบบ:

ระบบแสดงข้อความ error ว่า “**ผู้ใช้หรือรหัสผ่านไม่ถูกต้อง**”  
ไม่มีข้อมูลใดในฐานข้อมูลลูกค้าที่มีไดร์รับ token ที่ถูกต้อง

## บทที่ 6

### การวิเคราะห์ระบบด้วย AWS Well-Architected Framework

#### 6.1 Operational Excellence (ประสิทธิภาพการปฏิบัติงาน)

หมวด Operational Excellence ภายใต้ AWS Well-Architected Framework มุ่งเน้นการปรับปรุงกระบวนการดำเนินงานอย่างต่อเนื่อง (continuous improvement) เพื่อให้สามารถส่งมอบระบบที่มีคุณภาพ มีประสิทธิภาพในการจัดการ และสามารถรองรับการเปลี่ยนแปลงได้อย่างยืดหยุ่น

##### จุดแข็งของระบบ

###### 1. มีการใช้ระบบ Version Control (Git)

ระบบใช้ Git สำหรับการจัดการ source code อย่างเป็นระบบ ซึ่งช่วยให้สามารถติดตามประวัติการเปลี่ยนแปลงของโค้ด และลดความผิดพลาดในการพัฒนา

###### 2. มีการทดสอบเชิงฟังก์ชันก่อน Deploy

มีการเขียนและรัน Functional Test บางส่วนในระหว่างการพัฒนา เพื่อช่วยตรวจสอบความถูกต้องของฟีเจอร์หลักก่อนนำขึ้นใช้งานจริง

###### 3. การ Deploy เป็นกระบวนการแบบ Manual ที่ควบคุมได้

แม้ยังไม่ได้ใช้ CI/CD เต็มรูปแบบ แต่การ Deploy แต่ละครั้งมีการตรวจสอบ .env และความพร้อมของเซิร์ฟเวอร์ก่อนดำเนินการ

##### ข้อจำกัดและความเสี่ยงที่พบ

###### 1. ไม่มีระบบ Post-Incident Review หรือ Knowledge Management (OPS 11)

หากระบบเกิดข้อผิดพลาด ไม่มีแนวทางวิเคราะห์เหตุการณ์ย้อนหลังอย่างเป็นทางการ ซึ่งอาจส่งผลให้ปัญหาเดิมเกิดซ้ำ

###### 2. ไม่ได้จัดการ Configuration Management และ Patch Management อย่างมีระบบ (OPS 5)

ปัจจุบันยังไม่มีเครื่องมือหรือแนวทางเฉพาะในการจัดการคอนฟิกต่าง ๆ เช่น .env, ข้อมูลสิทธิ์ IAM หรือการอัปเดตเวอร์ชันของ dependency

###### 3. ไม่มี Observability Framework

ขาด Logging ที่รวมศูนย์ (Centralized Log), Metrics, Tracing หรือ Dashboard ที่ใช้ในการตรวจสอบสถานะระบบแบบเรียลไทม์

## แนวทางพัฒนาต่อ

เพื่อยกระดับประสิทธิภาพการปฏิบัติงานของระบบในอนาคต ควรพิจารณาดำเนินการดังนี้:

### 1. เพิ่ม Post-Incident Analysis

บันทึกเหตุการณ์ผิดพลาด และถอดบทเรียนเพื่อใช้ปรับปรุงการออกแบบระบบในระยะยาว

### 2. วางแผนใช้ระบบ CI/CD

เช่น GitHub Actions หรือ AWS CodePipeline เพื่อทำให้การ Build/Test/Deploy เป็นไปโดยอัตโนมัติ

### 3. นำระบบ Monitoring และ Alerting เข้ามาใช้

เช่น AWS CloudWatch หรือ Grafana + Prometheus เพื่อดูสถานะการทำงานของระบบอย่างต่อเนื่อง

### 4. ใช้ Configuration Management Tools (เช่น AWS SSM, dotenv vault)

เพื่อให้การจัดการคอนฟิกในหลายสภาพแวดล้อมมีความปลอดภัยและลดข้อผิดพลาดจากการมนุษย์

## 6.2 Security (ความปลอดภัย)

หมวด Security ใน AWS Well-Architected Framework เน้นที่การปกป้องข้อมูล ระบบ และทรัพยากรจากภัยคุกคามทั้งภายในและภายนอก ตลอดจนการสร้างความมั่นใจว่าระบบสามารถรักษาความลับ ความถูกต้อง และความพร้อมใช้งานของข้อมูลได้อย่างเหมาะสม

ระบบ ANTivity ได้เริ่มต้นพัฒนาในลักษณะที่คำนึงถึงความปลอดภัยในบางส่วน เช่น การจัดการผู้ใช้งานผ่าน AWS Cognito และการใช้ระบบยืนยันตัวตนก่อนเข้าถึงข้อมูล อย่างไรก็ตาม ยังมีหลายประเด็นที่ควรพัฒนาเพิ่มเติมเพื่อให้สอดคล้องกับแนวปฏิบัติที่ดีตามมาตรฐาน AWS

### จุดแข็งของระบบ

#### 1. ใช้ AWS Cognito สำหรับ Authentication และ Authorization

- ผู้ใช้งานต้องเข้าสู่ระบบผ่าน Cognito เพื่อรับ Token ก่อนเรียก API
- Token ถูกใช้ในการควบคุมการเข้าถึง (Access Control) ผ่าน Backend

#### 2. มีการใช้ระบบแยกบทบาทผู้ใช้ (Role-based Access)

- กำหนดสิทธิ์การเข้าถึงตามบทบาท ได้แก่ student, professor และ staff
- ทำให้แต่ละกลุ่มสามารถเข้าถึงเฉพาะส่วนที่เกี่ยวข้องได้

#### 3. มีการจัดการโค้ดที่ปลอดภัย

- มีการใช้ Git ในการควบคุมเวอร์ชันของซอฟต์แวร์
- มีการ Review โค้ดและจัดการ Build/Deployment อย่างมีระบบ

## ข้อจำกัดและความเสี่ยง

แม้ระบบมีการป้องกันพื้นฐานในระดับหนึ่ง แต่ยังพบความเสี่ยงในหลายด้านที่สำคัญ:

### 1. ไม่มีการเข้ารหัสข้อมูลที่จัดเก็บ (Data at Rest) (SEC 8)

ข้อมูลใน Amazon RDS และ S3 ยังไม่มีการเปิดใช้การเข้ารหัส (Encryption) ซึ่งเสี่ยงต่อการเข้าถึงโดยไม่ได้รับอนุญาต

### 2. ไม่มีการเข้ารหัสข้อมูลขณะส่ง (Data in Transit) (SEC 9)

ยังไม่มีการบังคับให้ใช้ HTTPS แบบเต็มรูปแบบในการเรียกใช้งาน API และ Frontend

### 3. ขาดการฝึกแนวทางความปลอดภัยในวงจรพัฒนา (DevSecOps) (SEC 11)

ยังไม่มีระบบตรวจสอบช่องโหว่ของ dependency หรือการทดสอบความปลอดภัยอัตโนมัติใน pipeline

### 4. ไม่มีการทำ Penetration Testing และ Security Assessment

- ยังไม่มีการประเมินความปลอดภัยโดยผู้เชี่ยวชาญหรือเครื่องมือวิเคราะห์ช่องโหว่
- ขาดการตั้งค่า alert หากมีการเข้าใช้งานระบบอย่างผิดปกติ

## แนวทางพัฒนาในอนาคต

เพื่อยกระดับความปลอดภัยของระบบ ควรพิจารณาดำเนินการดังนี้:

### 1. เปิดใช้ Encryption at Rest และ in Transit สำหรับ RDS และ S3

- ใช้ TLS/SSL สำหรับการเชื่อมต่อทั้งหมด
- ตั้งค่า RDS ให้ใช้การเข้ารหัสด้วย KMS key

### 2. นำแนวคิด DevSecOps เข้ามาใช้ในวงจรการพัฒนา

เช่น การใช้ Dependency Scanner (เช่น Snyk) และ Security Linter ใน GitHub Actions

### 3. เพิ่ม การตรวจสอบกิจกรรมที่ผิดปกติ (Audit Logging)

ใช้ AWS CloudTrail, Cognito Logs หรือ Integration กับ CloudWatch Logs

### 4. เพิ่ม การควบคุม IAM ที่เข้มงวด

จำกัดการเข้าถึง Bucket และ Database ตาม Principle of Least Privilege

### 5. นำระบบ 2FA/MFA มาใช้ในขั้นตอนสำคัญ

เช่น การจัดการกิจกรรมหรือข้อมูลอาจารย์

### 6.3 Reliability (ความน่าเชื่อถือ)

ความน่าเชื่อถือ (Reliability) หมายถึง ความสามารถของระบบในการทำงานตามหน้าที่ที่กำหนดได้อย่างสม่ำเสมอ โดยไม่เกิดการล้มเหลว และหากเกิดความผิดปกติขึ้น ระบบต้องสามารถฟื้นตัวกลับมาทำงานได้อย่างรวดเร็วและปลอดภัย AWS Well-Architected Framework ให้ความสำคัญกับการออกแบบระบบให้มีความทนทานต่อข้อผิดพลาด (fault-tolerant) มีความสามารถในการตรวจสอบปัญหา และมีแนวทางรองรับกรณีฉุกเฉิน

#### จุดแข็งของระบบ

##### 1. มีการจัดเก็บ Log สำหรับการวิเคราะห์ข้อมูล

- ระบบ Backend มีการบันทึก Log พื้นฐานเพื่อใช้ตรวจสอบข้อผิดพลาดที่เกิดขึ้นระหว่างการพัฒนาและใช้งาน
  - ข้อมูลจาก Log เหล่านี้สามารถนำไปใช้วิเคราะห์ปัญหาเชิงเทคนิคในภายหลังได้

##### 2. มีการเขียน Functional Test ระหว่างพัฒนา

- ใช้ในการตรวจสอบฟังก์ชันหลักของ API ให้ทำงานตามที่คาดหวัง เพื่อป้องกัน Regression Error
- ช่วยให้ระบบมีความมั่นคงก่อนนำเข้าใช้งานจริง

##### 3. ออกแบบระบบแยก Frontend และ Backend อย่างชัดเจน

ช่วยลดผลกระทบจากความผิดพลาดในแต่ละส่วน และเพิ่มความยืดหยุ่นในการจัดการปัญหาเฉพาะจุด ข้อจำกัดและความเสี่ยง

แม้จะมีพื้นฐานด้านความน่าเชื่อถือที่ดี แต่ยังพบข้อจำกัดที่ควรได้รับการพัฒนาเพิ่มเติมดังนี้:

##### 1. ขาดระบบ Monitoring และ Alerting แบบเรียลไทม์ (REL 6)

- ยังไม่มีการเก็บ Metrics ของ API, ความเร็วในการตอบสนอง หรือสถานะของ Service
- ไม่มีระบบแจ้งเตือน (Notification) เมื่อระบบมีผิดปกติ เช่น HTTP Error Rate เพิ่มสูงขึ้น

##### 2. ไม่มีการใช้ Runbook หรือ Resiliency Test (REL 8)

- ยังไม่มีเอกสาร Runbook ที่ชัดเจนสำหรับการกู้คืนระบบ
- ไม่มีการทดสอบความทนทานของระบบภายใต้สถานการณ์จำลอง เช่น เชิร์ฟเวอร์ล่ม

##### 3. ไม่ได้ทดสอบ Reliability ในเชิงโครงสร้าง (REL 12)

ยังไม่เคยมีการทดสอบด้วยแนวทางเช่น Chaos Engineering หรือ Game Day ซึ่งเป็นเทคนิคสำคัญ สำหรับระบบที่ต้องการรองรับการเติบโตในระดับ production

##### 4. ขาดแผน Disaster Recovery และ Fault Isolation

ระบบไม่มีการจัดทำสำรองข้อมูล (Backup Plan) ไม่มีแผนการฟื้นตัวจากเหตุการณ์ที่ส่งผลกระทบต่อทั้งระบบ เช่น EC2 หรือ RDS ไม่สามารถใช้งานได้

## แนวทางพัฒนาในอนาคต

### 1. เพื่อเพิ่มความน่าเชื่อถือของระบบ ควรพิจารณาดำเนินการดังนี้:

- นำ Monitoring และ Alerting Tools เข้ามาใช้
- เช่น AWS CloudWatch, Prometheus + Grafana หรือ Logtail

เพื่อติดตาม Availability, Error Rate และ Resource Usage แบบเรียลไทม์

### 2. เขียน Runbook สำหรับเหตุการณ์ฉุกเฉิน

เช่น วิธีแก้ไขเมื่อ Backend หยุดทำงาน หรือฐานข้อมูลขัดข้อง

### 3. ทดสอบความทนทานด้วย Chaos Engineering

เช่น จำลองการปิด EC2, ตัดการเชื่อมต่อกับ RDS เพื่อคุณภาพรวมของระบบ

### 4. ทำ Data Backup และวางแผน Disaster Recovery

การสำรองข้อมูล RDS อย่างสม่ำเสมอ และอาจใช้ Multi-AZ Deployment สำหรับความทนทานที่สูงขึ้น

## 6.4 Performance Efficiency (ประสิทธิภาพ)

ประสิทธิภาพการทำงาน (Performance Efficiency) คือความสามารถในการใช้ทรัพยากระบบที่เกิดประโยชน์สูงสุด เพื่อรับปริมาณงานและการเติบโตของระบบได้อย่างเหมาะสม โดยเน้นให้สามารถเลือกใช้บริการที่เหมาะสม ปรับขนาดได้ (Scalable) และตรวจวัดเพื่อปรับปรุงการใช้งานได้อย่างต่อเนื่อง

ระบบ ANTivity มีการวางแผนร่างแบบ Web-based Application ที่แยก Frontend และ Backend และเชื่อมโยงกับบริการหลักของ AWS ได้แก่ EC2, RDS, S3 และ Cognito ซึ่งถือว่าเป็นแนวทางที่เหมาะสมกับระบบในระดับเริ่มต้น แต่ยังมีหลายประเด็นสำคัญที่ต้องพัฒนาต่อในเชิงประสิทธิภาพ

### จุดแข็งของระบบ

#### 1. เลือกใช้บริการ Cloud ที่มีความยืดหยุ่นสูง

- ใช้ EC2 สำหรับทั้ง Frontend และ Backend ซึ่งสามารถปรับขนาด (Scale) ได้ในอนาคต
- ใช้ RDS PostgreSQL แบบ Free Tier ซึ่งเหมาะสมกับระบบขนาดเล็กในช่วงเริ่มต้น
- S3 ถูกใช้เก็บไฟล์รูปภาพ เพื่อไม่ให้ภาระตกลงกับเซิร์ฟเวอร์หลัก

#### 2. มีแนวโน้มการเรียนรู้การเลือกทรัพยากรให้เหมาะสม

ผู้พัฒนามีความเข้าใจพื้นฐานในการเลือกประเภทอินสแตนซ์ EC2 (เช่น t2.micro) เพื่อประหยัดทรัพยากรในระยะเริ่มต้น

#### 3. แยกส่วนระบบชัดเจนตามบทบาทการทำงาน

ส่งผลให้สามารถจัดสรรทรัพยากรและปรับปรุงเฉพาะจุดได้ในอนาคต เช่น scale backend ต่างหากจาก frontend

## ข้อจำกัดและความเสี่ยง

### 1. ขาดการใช้ข้อมูลเชิงวัดเพื่อปรับปรุงประสิทธิภาพ (PERF 1)

- ยังไม่มีการเก็บ Metrics เพื่อตรวจสอบว่า API หรือฐานข้อมูลได้มีภาระงานสูง
- ไม่ได้ใช้ข้อมูล Benchmark หรือ Policy เพื่อวางแผนการใช้ทรัพยากรให้เกิดประสิทธิภาพสูงสุด

### 2. ยังไม่มีการปรับแต่งระบบหรือใช้การ Optimize ที่ลึกซึ้น (PERF 2)

- ยังไม่มีการเลือกขนาด EC2 ตามภาระงานจริง หรือวิเคราะห์ Query ของฐานข้อมูลที่ใช้ปอย
- ยังไม่มีการเปิดใช้งาน Cache หรือ CDN (เช่น Amazon CloudFront) สำหรับรุปภาพหรือไฟล์หน้าเว็บ

### 3. ไม่มีการทำ Load Testing

- ระบบยังไม่เคยผ่านการทดสอบภายใต้สภาวะที่มีผู้ใช้งานจำนวนมากพร้อมกัน
- ไม่สามารถประเมินจุดคอขวด (Bottleneck) ของระบบในสถานการณ์จริงได้

## แนวทางพัฒนาในอนาคต

### 1. เก็บ Metrics และวิเคราะห์ทรัพยากร

- ใช้ AWS CloudWatch เพื่อตรวจสอบ CPU, Memory, IOPS และ Latency ของ EC2 และ RDS
- วิเคราะห์ Logs เพื่อปรับปรุง Query และโครงสร้างฐานข้อมูล

### 2. วางแผน Auto Scaling

พิจารณาใช้ Auto Scaling Group กับ EC2 ทั้ง Frontend/Backend เพื่อปรับขนาดอัตโนมัติตามปริมาณผู้ใช้

### 3. ใช้ Content Delivery Network (CDN)

เช่น Amazon CloudFront เพื่อเพิ่มความเร็วในการโหลดไฟล์รุปภาพจาก S3

### 4. ทำ Load Testing

ใช้เครื่องมือเช่น Apache JMeter, Artillery หรือ k6 เพื่อจำลองผู้ใช้งานจำนวนมาก และปรับปรุงประสิทธิภาพตามผลลัพธ์

### 5. เลือกใช้ Instance Types ที่เหมาะสม

เมื่อระบบขยายตัว ควรพิจารณาเปลี่ยนจาก t2.micro เป็นประเภทที่เหมาะสมกับลักษณะการทำงาน เช่น t3.medium, m5.large ฯลฯ

## 6.5 Cost Optimization (การจัดการต้นทุน)

Cost Optimization คือแนวทางในการลดต้นทุนของระบบโดยไม่ลดคุณภาพของบริการ โดย AWS Well-Architected Framework ส่งเสริมให้มีการวางแผนและบริหารจัดการต้นทุนอย่างเป็นระบบ ครอบคลุมการเลือกใช้บริการที่เหมาะสม การวิเคราะห์การใช้งานจริง และการพัฒนาโครงสร้างที่สามารถปรับลดหรือขยายได้ตามความจำเป็น

ระบบ ANTivity เป็นระบบที่พัฒนาในบริบทของโครงงานเพื่อการศึกษา โดยมีการใช้ AWS Resource ในระดับ Free Tier หรือใช้งานทรัพยากรอย่างจำกัด ทำให้ต้นทุนรวมยังไม่สูง อย่างไรก็ตาม หากมีแผนจะขยายระบบในอนาคต การจัดการต้นทุนจะมีบทบาทสำคัญต่อความยั่งยืนของระบบ

### จุดแข็งของระบบ

#### 1. เลือกใช้ทรัพยากรแบบ Free Tier และขั้นต่ำ

ใช้ EC2 ประเภท t2.micro และ Amazon RDS (PostgreSQL) แบบ Free Tier

เหมาะสมกับระบบที่ยังอยู่ในระยะทดลองและมีผู้ใช้จำกัด

#### 2. หลีกเลี่ยงการเปิดใช้งานบริการที่ไม่จำเป็น

ไม่มีการเปิดใช้บริการที่เกินความจำเป็น เช่น Load Balancer หรือ CloudFront ซึ่งช่วยลดค่าใช้จ่าย

#### 3. ใช้ AWS S3 สำหรับจัดเก็บไฟล์

ลดภาระพื้นที่และต้นทุนจากการเก็บรูปภาพไว้ในเครื่องเซิร์ฟเวอร์โดยตรง

### ข้อจำกัดและความเสี่ยง

#### 1. ไม่มีการประเมินทรัพยากรที่ใช้อย่างสม่ำเสมอ (COST 10)

ยังไม่มีการวิเคราะห์ Resource Usage เพื่อตรวจสอบว่าใช้งานเกินกว่าที่จำเป็นหรือไม่

#### 2. ไม่มีการใช้ Pricing Models ที่หลากหลาย

ปัจจุบันยังใช้แบบ On-Demand ทั้งหมด ไม่มีการพิจารณาใช้ Spot Instances หรือ Reserved Instances ที่สามารถช่วยลดต้นทุนได้ในระยะยาว

#### 3. ขาดระบบติดตามและแจ้งเตือนต้นทุน

ไม่มีการเปิดใช้งาน AWS Budgets หรือ Cost Explorer เพื่อตรวจสอบค่าใช้จ่ายแบบรายวัน/เดือน

#### 4. ไม่มีการแบ่ง Cost Attribution ตามฟีเจอร์หรือบริการ

ไม่สามารถวิเคราะห์ต้นทุนตามส่วนของระบบ เช่น Authentication, Storage หรือ Database แยกออกกัน

## แนวทางพัฒนาในอนาคต

### 1. เปิดใช้งาน AWS Budgets และ Cost Explorer

- ตั้งงบประมาณรายเดือน และแจ้งเตือนหากใกล้เกินงบ
- ใช้ข้อมูลจาก Cost Explorer ในการวางแผนบริการและวิเคราะห์ทรัพยากร

### 2. ประเมินและเลือกใช้ Instance แบบประหยัด

- หากระบบมีแนวโน้มใช้งานต่อเนื่อง ควรพิจารณาใช้ Reserved Instance
- หากระบบมีช่วงที่ไม่ใช้งาน อาจใช้ Auto Stop หรือ Spot Instance

### 3. วางแผนการนำ Cost Attribution

ใช้ Tag หรือชื่อ Resource เพื่อรับรู้ว่าแต่ละส่วนของระบบใช้งบเท่าไร เช่น frontend, backend, s3-storage

### 4. รีวิวและลบ Resource ที่ไม่ใช้งาน

ตรวจสอบ EC2, RDS และ Bucket ที่อาจเปิดทิ้งไว้โดยไม่ได้ใช้

### 5. ประเมิน Total Cost of Ownership (TCO) หากมีแผน Production

เพื่อเปรียบเทียบต้นทุนระหว่างบริการต่าง ๆ หรือการย้ายไปใช้แบบ Serverless

## 6.6 Sustainability (ความยั่งยืน)

Sustainability หรือ ความยั่งยืน เป็นหนึ่งในหลักการที่ถูกเพิ่มเติมเข้ามาใน AWS Well-Architected Framework เพื่อเน้นให้สถาปัตยกรรมระบบไอล์ฟไม่เพียงแต่ออกแบบให้มีประสิทธิภาพและความน่าเชื่อถือเท่านั้น แต่ยังต้องมีแนวทางการใช้ทรัพยากรที่เป็นมิตรต่อสิ่งแวดล้อม ลดคาร์บอนฟุตพريント และส่งเสริมการพัฒนาอย่างยั่งยืนในระยะยาว

แม้ว่าระบบ ANTivity จะยังอยู่ในระดับโครงงานของนิสิต และการใช้งานยังอยู่ในขอบเขตขนาดเล็ก แต่การตระหนักรู้เรื่องความยั่งยืนตั้งแต่ช่วงเริ่มต้นสามารถเป็นพื้นฐานที่ดีต่อการขยายระบบในอนาคต

### จุดแข็งของระบบ

#### 1. ใช้บริการ Cloud แทนการจัดการโครงสร้างพื้นฐานแบบ On-premises

- การเลือกใช้ AWS ช่วยลดความจำเป็นในการตั้งเซิร์ฟเวอร์ท้องถิ่น ซึ่งส่งผลโดยตรงต่อการลดการใช้พลังงานในสถานที่จริง (Data Center)

- AWS มีนโยบายใช้พลังงานหมุนเวียนในการให้บริการ ซึ่งช่วยลดการปล่อยก๊าซเรือนกระจกจากการอ้อม

## 2. ไม่มีการสร้างทรัพยากรเกินความจำเป็น

- ระบบใช้ทรัพยากรเพียงเท่าที่จำเป็น เช่น EC2 ประเภท t2.micro, RDS แบบ Free Tier และ S3 สำหรับจัดเก็บรูปภาพ

- การออกแบบเน้นประสิทธิภาพต่อทรัพยากร (Resource Efficiency) ในทุกขั้นตอน

### ข้อจำกัดและความเสี่ยง

#### 1. แนวปฏิบัติต้านความยั่งยืนยังอยู่นอกขอบเขตของโครงการ (Out of Scope)

- ยังไม่มีการวางแผนด้านความยั่งยืนที่ชัดเจน เช่น การติดตาม Carbon Footprint หรือการคำนวณ Power Utilization Effectiveness (PUE)

- ไม่มีการใช้บริการของ AWS ที่สนับสนุนการประเมินความยั่งยืนโดยตรง เช่น AWS Customer Carbon Footprint Tool

#### 2. ไม่มีการจัดเก็บหรือวิเคราะห์ข้อมูลด้านพลังงานหรือทรัพยากรที่ใช้งานจริง

ระบบยังไม่มีการวัดการใช้ CPU, Storage หรือ Network Bandwidth ในเชิงการลดพลังงาน  
แนวทางพัฒนาในอนาคต

##### 1. ใช้ AWS Regions ที่เน้นการใช้พลังงานสะอาด

เช่น us-west-2 (Oregon), eu-west-1 (Ireland) ซึ่ง AWS รายงานว่ามีการใช้พลังงานหมุนเวียนสูงกว่าค่าเฉลี่ย

##### 2. ใช้งาน Auto Scaling และ Serverless ในบางส่วนของระบบ

ลดการรัน EC2 ตลอดเวลา โดยย้ายส่วนที่ไม่จำเป็นไปยัง Lambda หรือ API Gateway  
ช่วยลดการใช้ทรัพยากรโดยไม่จำเป็นและส่งผลต่อพลังงานรวม

##### 3. วางแผนการจัดเก็บข้อมูลอย่างมีประสิทธิภาพ

ใช้ S3 Intelligent-Tiering สำหรับรูปภาพ เพื่อย้ายไฟล์เก่าไปเก็บในชั้นจัดเก็บที่ใช้พลังงานต่ำลง

##### 4. ศึกษาการใช้งาน AWS Sustainability Dashboard หรือ Carbon Footprint Tool

เพื่อตรวจสอบและประเมินผลกระทบต่อสิ่งแวดล้อมจากการใช้งานจริง

##### 5. สื่อสารแนวทางด้านความยั่งยืนต่อผู้ใช้งาน

สร้างความเข้าใจให้ทีมพัฒนาและผู้ใช้งานเกี่ยวกับการลดผลกระทบต่อสิ่งแวดล้อม

## บทที่ 7

### สรุปและแนวทางพัฒนาต่อ

#### 7.1 สรุปผลการดำเนินงานของโครงการ

โครงการระบบ ANTivity ได้รับการพัฒนาขึ้นเพื่อตอบโจทย์ด้านการติดตามและส่งเสริมการพัฒนาทักษะของนักศึกษาในระดับมหาวิทยาลัย ผ่านการมีส่วนร่วมในกิจกรรมเสริมหลักสูตร โดยระบบถูกออกแบบมาให้ครอบคลุมความต้องการของผู้ใช้งานหลัก 3 กลุ่ม ได้แก่ นักศึกษา, อาจารย์ที่ปรึกษา, และ เจ้าหน้าที่ผู้จัดกิจกรรม โดยมีเป้าหมายเพื่ออำนวยความสะดวกในการบริหารจัดการข้อมูลกิจกรรม วิเคราะห์ทักษะ และสนับสนุนการแนะนำแนวทางการพัฒนาตนเองของนักศึกษาอย่างเป็นระบบ

ในภาพรวม ระบบ ANTivity สามารถดำเนินการได้ครบถ้วนในแต่ละบทบาทผู้ใช้งาน โดยมีฟีเจอร์สำคัญที่ตอบสนองต่อความต้องการที่แท้จริง เช่น ระบบลงทะเบียนกิจกรรม, การยืนยันการเข้าร่วม, การบันทึกทักษะที่ได้รับ, การแสดงพัฒนาการในรูปแบบ Dashboard, ระบบอนุมัติและจัดการกิจกรรม รวมถึงฟังก์ชันการประเมินและให้ข้อเสนอแนะจากผู้ใช้งาน ซึ่งทั้งหมดนี้ถูกออกแบบและพัฒนาผ่านบริการของ AWS ได้แก่ Amazon EC2, RDS, S3 และ Cognito เพื่อรองรับการทำงานที่มีความยืดหยุ่น ปลอดภัย และสามารถขยายตัวได้ในอนาคต

ผลการทดสอบแบบ End-to-End ทั้ง 3 กรณีได้แสดงให้เห็นว่าระบบสามารถทำงานได้อย่างเสถียร มีความน่าเชื่อถือ และมีการจัดการกับข้อผิดพลาดอย่างเหมาะสม เช่น การตรวจสอบความถูกต้องของข้อมูลก่อนบันทึก การแสดงคำเตือนผู้ใช้อย่างชัดเจน และการป้องกันการบันทึกข้อมูลไม่ครบถ้วน ซึ่งล้วนเป็นปัจจัยสำคัญต่อความมั่นใจของผู้ใช้ในระบบ

นอกจากนี้ โครงการยังแสดงให้เห็นถึงความเข้าใจในหลักการของ AWS Well-Architected Framework ผ่านการวิเคราะห์จุดแข็งและจุดที่ยังต้องปรับปรุงในด้านต่าง ๆ เช่น ความปลอดภัย, ความน่าเชื่อถือ, ประสิทธิภาพ และการจัดการต้นทุน แม้ว่าบางแนวทางยังอยู่ในขอบเขตของโครงการในปัจจุบัน แต่ก็เป็นพื้นฐานที่ดีในการวางแผนพัฒนาและขยายระบบในอนาคต

โดยสรุป โครงการระบบ ANTivity นับว่าประสบความสำเร็จในการพัฒนา Web Application ที่สามารถนำไปประยุกต์ใช้ในสถานการณ์จริง เพื่อเสริมสร้างประสบการณ์การเรียนรู้ของนักศึกษา และยกระดับกระบวนการบริหารจัดการกิจกรรมในระดับมหาวิทยาลัยได้อย่างมีประสิทธิภาพ

## 7.2 ปัญหาและอุปสรรคที่พบ

ระหว่างการดำเนินโครงการพัฒนาและทดสอบระบบ ANTivity ทีมผู้พัฒนาได้เผชิญกับปัญหาและอุปสรรคหลายประการ ทั้งในด้านเทคนิค การออกแบบระบบ และการประสานการทำงานภายในกลุ่ม ซึ่งมีรายละเอียดที่สำคัญ ดังนี้:

### 1. ปัญหาด้านการเชื่อมต่อกับบริการ AWS

#### การตั้งค่า AWS Cognito และการจัดการกลุ่มผู้ใช้ (User Pool & Groups)

ในช่วงเริ่มต้นมีความสับสนเกี่ยวกับวิธีการกำหนดบทบาทผู้ใช้งาน (student, professor, staff) ผ่าน Cognito Group และการจัดการ Token ซึ่งส่งผลต่อการควบคุมสิทธิ์ในการเข้าถึง API ของแต่ละบทบาท

แนวทางแก้ไข: ศึกษาเอกสารของ AWS อย่างละเอียด พร้อมทดลองและตรวจสอบการเชื่อมโยงกับ Token ที่ Cognito ส่งกลับมาในการพิสูจน์ตัวตน

#### ปัญหาการตั้งค่า IAM และการเข้าถึง S3

ระบบไม่สามารถอัปโหลดหรืออ่านไฟล์ภาพจาก S3 ได้ในบางครั้ง เนื่องจากการตั้งค่า Bucket Policy หรือ CORS ที่ไม่ถูกต้อง

แนวทางแก้ไข: ปรับปรุง IAM Policy และ Bucket Permission ให้รองรับการเข้าถึงจาก frontend ได้อย่างถูกต้อง พร้อมตั้งค่า CORS ให้ครอบคลุม method ที่จำเป็น

### 2. ปัญหาด้านโครงสร้างฐานข้อมูล

#### การออกแบบ Schema ที่ซับซ้อนในช่วงแรก

ความสัมพันธ์ระหว่างตาราง เช่น student-activity-skill ต้องรองรับหลายเงื่อนไข เช่น การให้ทักษะหลังเข้าร่วม การประเมินผล การบันทึก feedback

แนวทางแก้ไข: ทำการปรับโครงสร้างฐานข้อมูลหลายรอบ โดยออกแบบให้รองรับเงื่อนไขที่ยืดหยุ่น และใช้ UUID สำหรับความปลอดภัยและการอ้างอิงระหว่างตาราง

#### ปัญหาการ Sync Schema อัตโนมัติ

เนื่องจากใช้ Deno + SQL Template แทน ORM สำเร็จรูป ทำให้ต้องจัดการ migration และคำสั่ง CREATE TABLE ด้วยตนเอง

แนวทางแก้ไข: ใช้แนวทาง incremental schema พร้อมคำสั่ง IF NOT EXISTS เพื่อป้องกันการเกิด error ในการสร้างตารางซ้ำ

### 3. ข้อจำกัดจาก AWS Learner Lab

#### ข้อจำกัดด้านเวลาใช้งานต่อ session

ในช่วงพัฒนาโครงการ ระบบต้องใช้เวลาในการ deploy และทดสอบหลายรอบ แต่ AWS Learner Lab มีการกำหนดระยะเวลา session ที่จำกัด (4 ชั่วโมงต่อครั้ง) เมื่อ session หมด ระบบจะถูกปิดอัตโนมัติ ทำให้ต้องเริ่มกระบวนการใหม่อีกครั้ง ส่งผลให้การทดสอบระบบขาดความต่อเนื่อง

แนวทางแก้ไข: วางแผนการใช้งานให้เหมาะสมกับเวลาที่ระบบเปิด session พร้อมจัดลำดับความสำคัญของงานในแต่ละรอบ

#### ข้อจำกัดของ IAM Role และ Policy

ในช่วงเริ่มต้นที่มุ่งพัฒนาข้อจำกัดในการจัดการ IAM Role และ Policy ภายใน AWS Learner Lab ซึ่งไม่สามารถปรับแต่งสิทธิ์ได้อย่างอิสระ เช่น ไม่สามารถสร้าง policy แบบ custom หรือกำหนดสิทธิ์เฉพาะเจาะจงให้กับบริการบางประเภท ทำให้บางครั้งการเข้าถึงบริการหรือทรัพยากรบางอย่างถูกจำกัดจนไม่สามารถทดสอบได้ครบถ้วน

แนวทางแก้ไข: ศึกษาข้อจำกัดของ IAM ที่กำหนดไว้ใน Learner Lab และเลือกใช้บริการที่อยู่ภายใต้สิทธิ์ที่อนุญาตเท่านั้น รวมถึงปรับเปลี่ยนแนวทางการพัฒนาให้เหมาะสมกับขอบเขตที่สามารถใช้งานได้จริง

### 4. ปัญหาในการแบ่งงานและการทำงานร่วมกันในทีม

#### การแบ่งหน้าที่ที่ไม่ชัดเจน

ในช่วงต้นของการ พัฒนา สมาชิกยังไม่เข้าใจบทบาทของตนเองอย่างชัดเจน ทำให้เกิดความซ้ำซ้อนในการทำงาน เช่น หลายคนพัฒนางานเดียวกัน หรือบางงานไม่มีผู้รับผิดชอบ

แนวทางแก้ไข: มีการประชุมกำหนดบทบาทและขอบเขตงานของแต่ละคนอย่างชัดเจน

#### การติดตามความคืบหน้าของงานในทีม

ขาดระบบหรือตารางเวลาชัดเจนในการติดตามความก้าวหน้า ผลงาน ทำให้เกิดความล่าช้า และไม่สามารถเชื่อมโยงกับงานอื่นได้ทันท่วงที

แนวทางแก้ไข: จัด stand-up meeting เป็นประจำ และใช้เครื่องมือติดตามงานเพื่อให้ทุกคนเห็นภาพรวม

#### การสื่อสารภายในทีมไม่เต็อเนื่อง

สมาชิกบางคนไม่ได้อัปเดตข้อมูลหรือแจ้งปัญหาที่พบ ทำให้ทีมไม่สามารถช่วยเหลือหรือปรับแผนได้ทัน

แนวทางแก้ไข: กำหนดช่องทางการสื่อสารหลักของทีม Discord พร้อมกำหนดให้มีการอัปเดตงานรายวัน หรือรายสัปดาห์

### 7.3 แนวทางการพัฒนาต่อยอดในอนาคต

แม้ว่าโครงการ ANTivity จะสามารถตอบโจทย์การติดตามและวิเคราะห์ทักษะของนักศึกษาได้ในระดับพื้นฐานอย่างครบถ้วนแล้ว แต่อย่างไรก็ตาม ยังมีโอกาสอีกมากในการพัฒนาและต่อยอดระบบให้สามารถรองรับการใช้งานในระดับที่กว้างขึ้น มีความยืดหยุ่นและอัจฉริยะยิ่งขึ้น เพื่อรับความต้องการที่เปลี่ยนแปลงและสนับสนุนการเรียนรู้ตลอดชีวิตของผู้ใช้งานอย่างแท้จริง โดยมีแนวทางที่ควรพิจารณาในอนาคต ดังต่อไปนี้:

#### 1. ระบบแนะนำกิจกรรมโดยอิงตามความสามารถ (Personalized Activity Recommendation)

พัฒนาระบบแนะนำกิจกรรมเสริมทักษะโดยใช้ Machine Learning (ML) หรือ Rule-based engine เพื่อวิเคราะห์/perfle ทักษะของนักศึกษาแต่ละคน เปรียบเทียบกับเป้าหมายของหลักสูตร และแนะนำกิจกรรมที่เหมาะสม เช่นรายบุคคล ซึ่งจะช่วยให้นักศึกษาสามารถวางแผนพัฒนาทักษะได้อย่างมีเป้าหมาย ชัดเจน และสอดคล้องกับอาชีพในอนาคต

#### 2. การแสดงผลทักษะแบบแผนภาพ (Skill Map Visualization)

เพิ่มฟีเจอร์แผนภาพ Skill Map แบบ Interactive เพื่อแสดงความเชื่อมโยงระหว่างกิจกรรมที่เข้าร่วม ทักษะที่ได้รับ และทักษะที่ยังขาด พร้อมวางแผนเป้าหมายการพัฒนาในระยะยาว วิธีนี้ช่วยให้นักศึกษาเห็นภาพรวมพัฒนาการของตนเอง และกระตุ้นให้เกิดการวางแผนเรียนรู้ด้วยตนเองได้อย่างมีประสิทธิภาพ

#### 3. ระบบยืนยันการเข้าร่วมด้วย QR Code

พัฒนาระบบ QR Code Check-in ที่นักศึกษาสามารถสแกนเพื่อยืนยันการเข้าร่วมกิจกรรมได้หน้างานจริง ช่วยลดข้อผิดพลาดจากการบันทึกมือ ลดการทุจริต และเพิ่มความถูกต้องของข้อมูล พร้อมสามารถบันทึกเวลาเข้า-ออกกิจกรรมได้อย่างแม่นยำ

#### 4. การออกรายงานการพัฒนาทักษะแบบ PDF

เพิ่มฟีเจอร์ Export รายงานผลทักษะ ในรูปแบบไฟล์ PDF สำหรับทั้งนักศึกษา อาจารย์ที่ปรึกษา และเจ้าหน้าที่ โดยรายงานจะสรุปผลการเข้าร่วมกิจกรรม พัฒนาการด้านทักษะ และเปรียบเทียบกับเกณฑ์ของหลักสูตร พร้อมแสดงข้อมูลในรูปแบบตารางและกราฟ เพื่อใช้ในการแนะนำ หรือประกอบ Portfolio

#### 5. รองรับกิจกรรมนอกระบบ (External Activity Integration)

เปิดให้ระบบสามารถรองรับ กิจกรรมภายนอกมหาวิทยาลัย เช่น องค์กรภาคเอกชน มูลนิธิ หรือชมรมต่าง ๆ โดยมีระบบตรวจสอบคุณภาพกิจกรรมและการรับรองทักษะ เพื่อเพิ่มความหลากหลายและเปิดโอกาสให้นักศึกษาได้เรียนรู้นอกห้องเรียนมากยิ่งขึ้น

#### 6. การวิเคราะห์สถิติในระดับองค์กร (Institutional Analytics)

พัฒนา Dashboard วิเคราะห์ แนวโน้มการพัฒนาทักษะรายคณะ/รายสาขา เพื่อใช้ในการวางแผนการจัดกิจกรรมเชิงกลยุทธ์ของแต่ละหน่วยงานในมหาวิทยาลัย รวมถึงการประเมินผลลัพธ์การเรียนรู้ของหลักสูตรอย่างเป็นระบบ (Outcome-Based Education: OBE)

## 7. ระบบรองรับ Multi-language และ Accessibility

เพื่อขยายการใช้งานระบบในระดับที่กว้างขึ้น ควรพัฒนาระบบให้รองรับ หลายภาษา (Multi-language) และสามารถเข้าถึงได้สำหรับผู้ใช้ที่มีข้อจำกัด เช่น ผู้พิการทางสายตา (ผ่าน WAI-ARIA หรือ voice support) ซึ่งจะช่วยให้ระบบเป็นมิตรและเท่าเทียมสำหรับผู้ใช้งานทุกกลุ่ม

## 8. ระบบประเมินทักษะอัตโนมัติผ่าน AI (AI-based Skill Assessment)

เพิ่มฟีเจอร์ การประเมินทักษะอัตโนมัติ โดยใช้เทคนิค NLP หรือ AI Models วิเคราะห์สิ่งที่นักศึกษาทำหรือเขียน (เช่น บันทึกการสะท้อนหลังร่วมกิจกรรม การส่งแบบฝึกหัด หรือไฟล์ผลงาน) เพื่อประเมินระดับทักษะที่ได้รับโดยอัตโนมัติ ลดภาระอาจารย์ และให้ผลลัพธ์ที่สม่ำเสมอ พร้อมเสริมด้วย Feedback เชิงคุณภาพที่เป็นกลาง

ตัวอย่างเทคโนโลยี:

ใช้ GPT หรือ BERT model สำหรับวิเคราะห์ข้อความ

ใช้ Rubric-based AI Grading สำหรับประเมินความครบถ้วนของ soft skill

ใช้ Object Detection/Code Analysis สำหรับประเมิน hard skill เช่น coding หรือ visual artifacts

## 9. ระบบการแจ้งเตือน

มีระบบการแจ้งเตือนต่างๆ เช่น การแจ้งเตือนเมื่อใกล้ถึงวันหรือเวลาเข้าร่วมกิจกรรม การแจ้งเตือนหากสกิลที่มีอยู่ยังไม่ถึงเกณฑ์ของหลักสูตร การแจ้งเตือนเมื่อลองทะเบียนเข้าร่วมกิจกรรมแล้ว เป็นต้น โดยที่ผู้ใช้สามารถตั้งค่าการแจ้งเตือนต่างๆ เองได้ตามความต้องการ

## 10. ระบบการตั้งเป้าหมายของนักศึกษา

นักศึกษาสามารถตั้งเป้าหมายสกิลที่ตนเองต้องการเพิ่มเติมนอกเหนือจากที่หลักสูตรมีอยู่ได้ เพื่อให้นักศึกษาสามารถเข้าร่วมกิจกรรมที่ตนเองสนใจจริงๆ และเป็นการให้นักศึกษาเห็นว่าเป้าหมายที่ตั้งไว้มีความคืบหน้าแค่ไหนผ่านสกิลที่นักศึกษาได้รับ

## 11. พัฒนาต่อให้เป็นแอปพลิเคชัน

ปัจจุบันโครงการของเราเป็นการทำเว็บไซต์ โดยที่มี AWS เป็นตัวช่วย แต่หากต้องนำไปต่อยอดเพื่อใช้จริง ควรจะพัฒนาเป็นแอปพลิเคชันสำหรับมือถือเพื่อความสะดวกสบายในการใช้งานของผู้ใช้ และเป็นการเพิ่มตัวเลือกให้ผู้ใช้ที่ไม่ต้องการเข้าสู่หน้าเว็บไซต์

## 7.4 ข้อเสนอแนะเพื่อการปรับปรุงระบบ

แม้ระบบ ANTivity ได้รับการออกแบบและพัฒนาให้สามารถใช้งานได้อย่างครอบคลุมตามวัตถุประสงค์หลักของโครงการ แต่จากการวิเคราะห์การใช้งานจริง รวมถึงผลตอบรับจากผู้ใช้งานในแต่ละบทบาท ยังพบว่ามีประเด็นที่สามารถปรับปรุงได้เพิ่มเติม เพื่อเพิ่มประสิทธิภาพ ความยืดหยุ่น และประสบการณ์ผู้ใช้งาน ดังนี้

### 1. ปรับปรุงความเสถียรของระบบ (System Stability)

ควรมีการติดตั้งระบบ Monitoring และ Logging แบบคร่าวๆ เพื่อให้สามารถติดตามสถานะการทำงานของ Backend, Database และ EC2 ได้แบบเรียลไทม์ พร้อมมีระบบแจ้งเตือนเมื่อเกิดความผิดพลาด เช่น API ล่ม หรือฐานข้อมูลไม่ตอบสนอง ซึ่งจะช่วยให้สามารถแก้ไขปัญหาได้รวดเร็วขึ้น

### 2. ปรับปรุงการยืนยันตัวตนและสิทธิ์ของผู้ใช้งาน (Access Control)

แม้ระบบใช้ AWS Cognito จัดการบัญชีผู้ใช้ แต่ยังสามารถเพิ่มระบบ UI สำหรับการจัดการ Group และ Role-Based Access ให้ใช้งานง่ายขึ้น เช่น การกำหนดสิทธิ์เฉพาะเมนูจากหน้าแดชบอร์ด โดยไม่ต้องแก้ไขโค้ด

### 3. ปรับปรุงประสบการณ์ผู้ใช้ (User Experience)

ควรมีการเก็บ Feedback จากผู้ใช้งานจริง (นักศึกษา อาจารย์ เจ้าหน้าที่) เพื่อปรับปรุงหน้า UI/UX ให้เหมาะสม ตัวอย่างเช่น

- การแสดงสถานะกิจกรรม (อนุญาต/ไม่อนุญาต) ให้ชัดเจน
- ปรับการจัดกลุ่มทักษะให้อ่านง่าย และใช้ icon แทนข้อความบางส่วน
- เพิ่มระบบแจ้งเตือนผ่าน email หรือ notification เมื่อมีกิจกรรมใหม่หรือสถานะกิจกรรมเปลี่ยนแปลง

### 4. ปรับปรุงระบบการประเมินและให้ทักษะ

ในขั้นตอนการให้ทักษะแก่นักศึกษาหลังจบกิจกรรม ควรเพิ่มระบบช่วยเตือนเมื่อข้อมูลยังไม่ครบ เช่น:

- ยังไม่ได้ประเมินกิจกรรม
- ยังไม่ได้เลือกทักษะ
- ยังไม่มีข้อมูลการยืนยันการเข้าร่วม

รวมถึงเพิ่มระบบ suggestion ทักษะอัตโนมัติ เป็นต้นจากประเภทกิจกรรมเพื่อช่วยเจ้าหน้าที่ให้ทำงานเร็วขึ้น

### 5. เพิ่มระบบ Backup และ Recovery สำหรับฐานข้อมูล

เพื่อความมั่นคงของข้อมูลระบบ ควรตั้งค่า Automated Snapshot สำหรับ Amazon RDS และมีคู่มือหรือ script การ restore ระบบในการกรณีเกิดเหตุการณ์ฉุกเฉิน (Disaster Recovery Plan)

### 6. เสริมระบบการประเมินคุณภาพกิจกรรม (Activity Quality Feedback)

ควรเพิ่มการวิเคราะห์ข้อมูล feedback จากนักศึกษาที่เข้าร่วม เช่น การให้คะแนนผู้จัด การวิเคราะห์ความสนใจในหัวข้อกิจกรรม และแนวโน้มการเข้าร่วมกิจกรรม เพื่อเป็นข้อมูลเชิงสถิติให้เจ้าหน้าที่ใช้ปรับปรุงในรอบต่อไป

## 7. เตรียมความพร้อมสำหรับการขยายระบบ (Scalability Plan)

หากระบบจะนำไปใช้ในระดับคณะหรือมหาวิทยาลัยอย่างเต็มรูปแบบ ควรมีการวางแผนการใช้งานร่วมกับ Load Balancer, Auto Scaling และระบบ CI/CD เพื่อลด Downtime และรองรับผู้ใช้งานพร้อมกันจำนวนมาก

## 8. เพิ่มระบบ Export ข้อมูลและการเข้าถึงด้วย API ภายนอก

เพื่อให้ระบบมีความยืดหยุ่น ควรเพิ่มความสามารถในการ ส่งออกข้อมูลกิจกรรม ทักษะ และประวัตินักศึกษา เป็น Excel หรือ PDF และเปิดให้สามารถ เข้าถึงผ่าน REST API สำหรับการเชื่อมต่อกับระบบอื่น เช่น ระบบการศึกษา หรือ ระบบลงทะเบียนกลางของมหาวิทยาลัย

ភាគីនវក

ภาคผนวก ก: URL Git Repository และลิงก์คลิปวิดีโอสาธิต

เพื่อแสดงหลักฐานการพัฒนาและสาธิตการทำงานของระบบ ANTivity อย่างครบถ้วน รายละเอียดของแหล่งที่เก็บโค้ดและวิดีโอสาธิตการใช้งานมีดังนี้:

## 1. Git Repository

ระบบถูกพัฒนาและจัดการเวอร์ชันผ่าน GitHub โดยมีการแบ่งโครงสร้างเป็น Backend และ Frontend

URL GitHub Repository (Public):

<https://github.com/Somxhai/cloud-project>

## 2. คลิปวิดีโอสาธิตการทำงานของระบบ

คลิปวิดีโอดังตัวอย่างการใช้งานระบบในบทบาทต่าง ๆ ได้แก่ นักศึกษา อาจารย์ที่ปรึกษา และเจ้าหน้าที่ พร้อมแสดงให้เห็นการทำงานร่วมกับ AWS Services ที่ใช้จริงระหว่างรันระบบ

ลิงก์คลิปวิดีโอ:

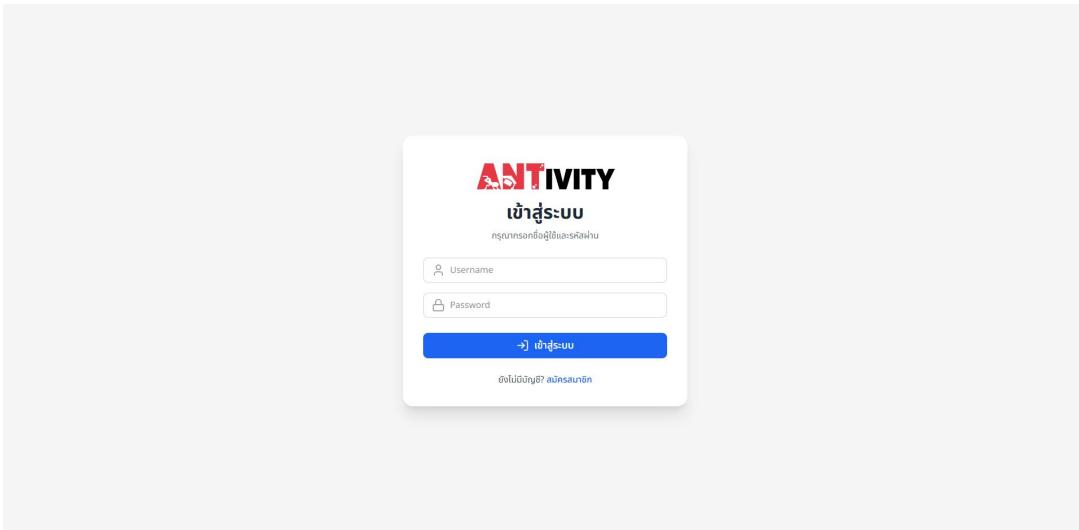
[https://tui pied-my.sharepoint.com/:v/g/personal/thanapoom\\_pol\\_dome\\_tu\\_ac\\_th/EYgZG1ieVh5JuyHrvscTpAcBtMubDinWfMRev1y8swdDsA?e=z6chud](https://tui pied-my.sharepoint.com/:v/g/personal/thanapoom_pol_dome_tu_ac_th/EYgZG1ieVh5JuyHrvscTpAcBtMubDinWfMRev1y8swdDsA?e=z6chud)

## ภาคผนวก ข: ภาพตัวอย่างส่วนติดต่อผู้ใช้ (UI/UX) จากระบบจริง

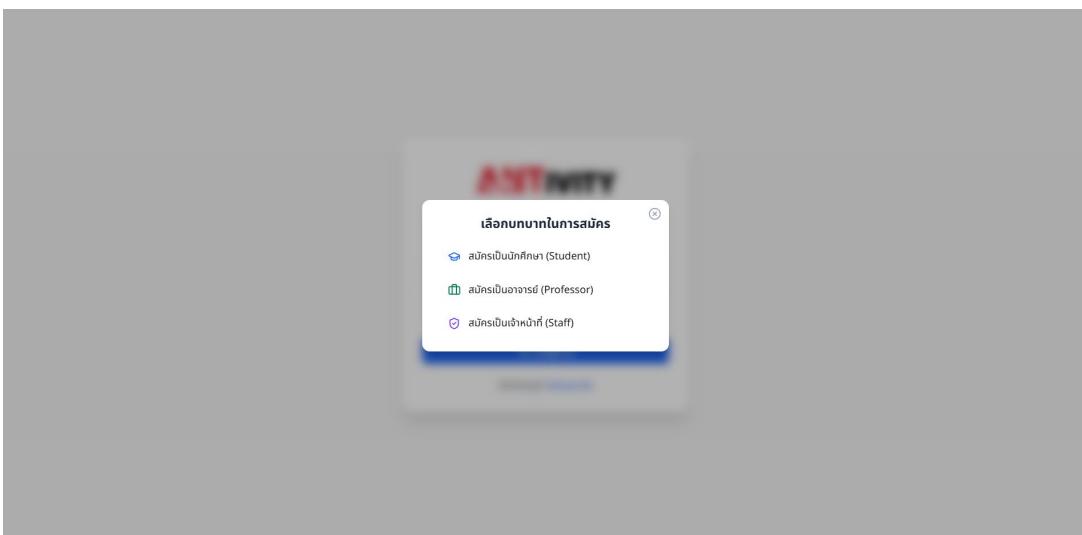
ในภาคผนวกนี้ เป็นการรวบรวมภาพหน้าจอ (Screenshot) จากระบบจริงที่พัฒนาขึ้น เพื่อแสดงให้เห็นตัวอย่าง ส่วนติดต่อผู้ใช้งาน (User Interface) และลำดับขั้นตอนการใช้งานของระบบในบทบาทต่าง ๆ ได้แก่ นักศึกษา อาจารย์ที่ปรึกษา และเจ้าหน้าที่ผู้จัดกิจกรรม โดยภาพทั้งหมดแสดงผลจากการใช้งานจริงบนระบบ ANTivity ที่มีการทดสอบแบบ End-to-End แล้ว

### 1. ใช้ร่วม

#### 1.1 หน้าเข้าสู่ระบบ



#### 1.2 หน้าต่างเมื่อกดสมัครสมาชิก ซึ่งจะแสดงบทบาทในการสมัครให้เลือก



## 2. เจ้าหน้าที่ผู้จัดกิจกรรม

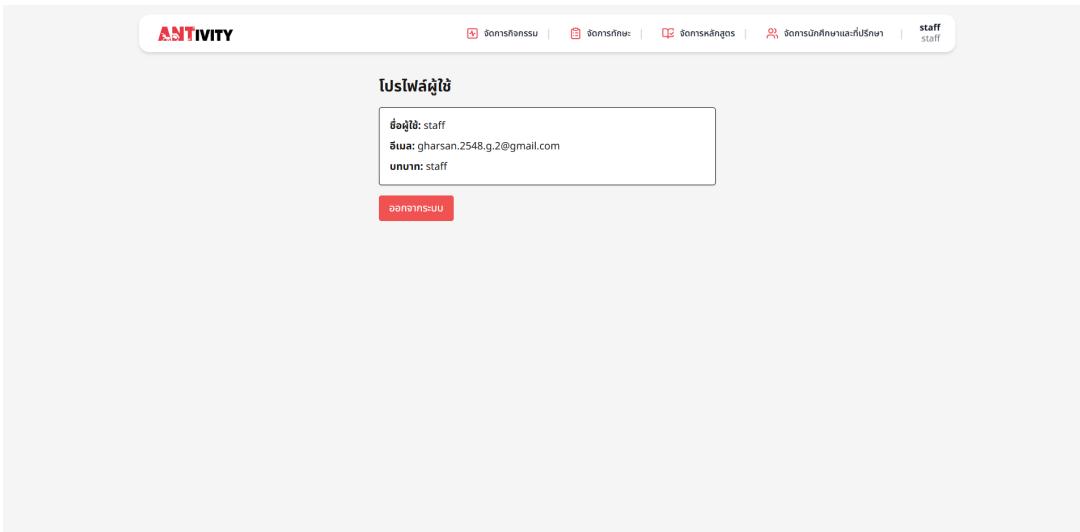
### 2.1 หน้าสมัครสมาชิกเจ้าหน้าที่

The screenshot shows a registration form for 'Staff' users. The title 'ANTIVITY' is at the top in red. Below it is the text 'สมัครสมาชิกเจ้าหน้าที่ (Staff)'. There are three input fields: 'Username' (with a user icon), 'Email' (with an envelope icon), and 'Password' (with a lock icon). A blue button labeled '→ ต่อไป (ต่อไป)' is below the fields. At the bottom, there is a link 'มีบัญชีอยู่แล้ว? เข้าสู่ระบบ'.

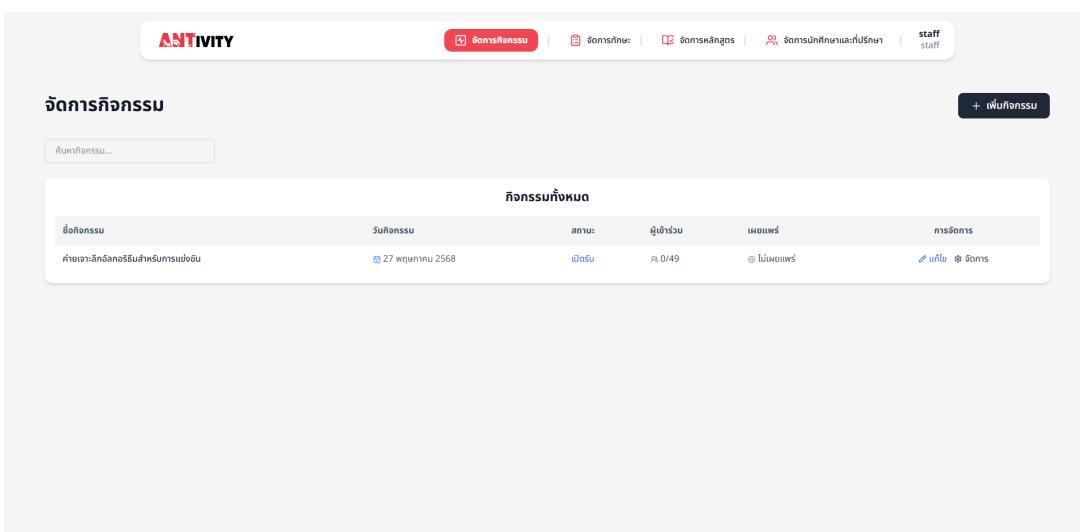
### 2.2 หน้ากรอกรหัสสีน้ำเงินผ่านอีเมลของเจ้าหน้าที่

The screenshot shows a password reset form for 'Staff' users. The title 'ANTIVITY' is at the top in red. Below it is the text 'สมัครสมาชิกเจ้าหน้าที่ (Staff)'. There is one input field for 'رمزรหัสที่ได้รับทางอีเมล' (Email received) with a mail icon. A green button labeled 'รีเซ็ตรหัส' (Reset Password) is below the field.

## 2.3 หน้าໂປຣໄຟລ໌ໃຫ້ອານເຈົ້າໜ້າທີ່



## 2.4 หน้าຈັດການກົງກະຽມທີ່ໜ້າ



## 2.5 หน้าเพิ่มกิจกรรมใหม่

หน้าเพิ่มกิจกรรมใหม่

ข้อมูลพื้นฐาน

ชื่อกิจกรรม  
รายละเอียด  
รายละเอียดการจัดกิจกรรม  
เมื่อไฟล์ ไม่มีเมื่อไฟล์

กำหนดการ & สถานที่

วัน/เดือนปีปัจจุบัน  
สถานที่  
จังหวัดผู้เข้าร่วมอยู่

สถานะ & การเผยแพร่

สถานะ  
เผยแพร่ทั่วไป

กำหนดการ & ให้รับ

การต้องการ  
 การต้องการ (soft)  
 การต้องการหนัก (hard)  
 การต้องการอย่างเดือด (hard)

ระดับ 3  
ระดับ 1  
ระดับ 2  
ระดับ 3  
ระดับ 4  
ระดับ 5

การดำเนินการ (soft)  
การติดต่อทางโทรศัพท์ (soft)

บันทึกกิจกรรม

## 2.6 หน้าจัดการกิจกรรม

The screenshot shows the 'Competitive Programming' activity setup page. At the top, there are tabs for 'จัดการกิจกรรม' (Activity Management), 'จัดการกิจ忙' (Activity Status), 'จัดการหลักสูตร' (Curriculum Management), 'จัดการนักกีฬาและทีมรักษา' (Athlete and Medical Team Management), and 'staff'. Below the tabs, it says 'ค่ายฯลฯ กิจกรรมสำหรับการแข่งขัน' (Camp activities for competition) and 'กิจกรรมที่ยังไม่ได้เปิดเผย' (Unpublished activities). The main content area includes sections for 'ข้อมูลกิจกรรม' (Activity Information), 'การตั้งค่า' (Setting), and a large blue button labeled 'บันทึกการเปลี่ยนแปลง' (Save changes).

This screenshot shows the same activity setup page with more detailed configurations. It includes sections for 'จำนวนผู้เข้าร่วม' (Number of participants), 'กำหนดการ' (Schedule), and 'รายละเอียด' (Details). The 'รายละเอียด' section contains dropdown menus for 'จำนวนผู้เข้าร่วม' (0), 'จำนวนผู้ดูแล' (0), 'จำนวนผู้สอน' (0), 'จำนวนครัวเรือน' (0), and 'จำนวนผู้ช่วย' (0). There is also a 'โหลดไฟล์' (Load file) button.

This screenshot shows the activity setup page with advanced settings. It includes sections for 'การกำหนดระดับความยาก' (Difficulty level setting) and 'การตั้งค่าของผู้ดูแล' (Administrator settings). The 'การกำหนดระดับความยาก' section lists four difficulty levels: 'soft' (ระดับ 2), 'hard' (ระดับ 2), 'medium' (ระดับ 2), and 'soft' (ระดับ 3). Each level has a dropdown menu with the text 'บันทึกเพื่อปรับ' (Save to adjust). The 'การตั้งค่าของผู้ดูแล' section has a single button labeled 'บันทึกการเปลี่ยนแปลง' (Save changes).

The screenshot shows a course page on the Activity application. The course title is 'ค่ายเจาะลึกอัลกอริธึมสำหรับการแข่งขัน' (Competitive Programming). It is marked as 'Unpublished' and has a status of 'staff'. Below the title, there are tabs for 'ห้องเรียน' (Classroom), 'จัดการกิจกรรม' (Activity Management), 'จัดการหลักสูตร' (Curriculum Management), 'จัดการนักศึกษาและบุคลากร' (Student and Faculty Management), and 'staff staff'. There is also a 'staff' button at the top right.

## 2.7 หน้าจัดการทักษะ

The screenshot shows the 'จัดการทักษะ' (Skill Management) section of the Activity application. It displays various skills categorized into soft and hard skills. The skills listed are:

- การสื่อสาร** (Communication): A soft skill related to communication and collaboration.
- การคิดเชิงวิพากษ์** (Critical Thinking): A soft skill related to problem-solving and critical analysis.
- การทักษะบันทึก** (Teamwork): A soft skill related to working effectively in teams.
- การจัดการฐานข้อมูล** (Database Management): A hard skill related to managing databases like MySQL and PostgreSQL.
- การพัฒนาเว็บไซต์** (Web Development): A hard skill related to web development using HTML, CSS, JavaScript, and other technologies.

## 2.8 ส่วนการเพิ่มทักษะและการแก้ไขรายละเอียดทักษะ

The screenshot shows the 'เพิ่มทักษะ' (Add Skill) dialog box and a detailed view of the 'การสื่อสาร' (Communication) skill.

**เพิ่มทักษะ:**

- ชื่อทักษะ: ภาษาไทย
- ชื่อภาษาอังกฤษ: Thai Language
- ประเภท: Soft Skill
- สถานะ: Active

**การสื่อสาร** (Communication)

ความสามารถในการสื่อสารภาษาไทยและภาษาอังกฤษได้ดี สามารถเข้าใจและอธิบายได้โดยรอบ

soft Active

## 2.9 หน้าจัดการหลักสูตรทั้งหมด

The screenshot shows the 'List of Courses' page of the ANTIVITY platform. At the top, there is a navigation bar with icons for 'List of courses', 'List of subjects', 'List of modules', 'List of teachers', and 'staff'. Below the navigation bar, the title 'รายชื่อหลักสูตร' (List of Courses) is displayed. A search bar with placeholder text 'ค้นหาหลักสูตร' (Search course) and three small icons is present. The main area contains a table with columns for 'ชื่อหลักสูตร\*', 'จำนวนครุภัณฑ์', and 'จำนวนผู้สอน'. There are also buttons for 'เพิ่มหลักสูตรใหม่' (Add new course) and 'ลบ' (Delete).

## 2.10 หน้าเพิ่มหลักสูตรใหม่

The screenshot shows the 'Add New Course' form of the ANTIVITY platform. At the top, there is a navigation bar with icons for 'List of courses', 'List of subjects', 'List of modules', 'List of teachers', and 'staff'. Below the navigation bar, the title 'เพิ่มหลักสูตรใหม่' (Add new course) is displayed. The form has two input fields: 'ชื่อหลักสูตร\*' (Course name) with placeholder text 'ใส่ชื่อหลักสูตรเพิ่งต้อง' (Enter course name) and 'จำนวนครุภัณฑ์' (Number of materials) with placeholder text 'รายละเอียดเพิ่มเติม (ไม่มีข้อบกพร่อง)' (Additional details (no errors)). At the bottom right, there is a blue button labeled '+ สร้างหลักสูตร' (Create course).

## 2.11 หน้าจัดการหลักสูตร

The screenshot shows the ANTIVITY platform's management interface for courses. At the top, there are four tabs: 'Management' (selected), 'Teacher Management', 'Student Management', and 'Staff Management'. Below the tabs, the title 'Course Management' is displayed. A blue button labeled 'Add Course' is visible. The main content area is titled 'Course Information' and contains sections for 'Course Name' (input field), 'Course Description' (input field), 'Course Type' (dropdown menu with 'Theory'), 'Course Duration' (input field), 'Course Status' (dropdown menu with 'Active'), and 'Course Status' (checkbox). There is also a 'Save' button.

## 2.12 ส่วนจัดการทักษะที่ต้องการสำหรับหลักสูตร

The screenshot shows the ANTIVITY platform's skills management interface. At the top, there are four tabs: 'Management' (selected), 'Teacher Management', 'Student Management', and 'Staff Management'. Below the tabs, the title 'Skill Management' is displayed. A blue button labeled 'Add Skill' is visible. The main content area is titled 'Skill Information' and contains sections for 'Skill Name' (input field), 'Skill Description' (input field), 'Skill Level' (dropdown menu with 'Easy'), and 'Skill Status' (checkbox). There is also a 'Save' button.

The screenshot shows the ANTIVITY platform's course skills management interface. At the top, there are four tabs: 'Management' (selected), 'Teacher Management', 'Student Management', and 'Staff Management'. Below the tabs, the title 'Course Skills Management' is displayed. A blue button labeled 'Add Course Skill' is visible. The main content area is titled 'Course Skills Information' and contains sections for 'Course Name' (input field), 'Skill Name' (input field), 'Skill Level' (dropdown menu with 'Easy'), and 'Skill Status' (checkbox). There is also a 'Save' button.

## 2.13 หน้าจัดการกิจกรรมส่วนผู้เข้าร่วมเมื่อมีคำขอเข้าร่วมกิจกรรม

The screenshot shows the 'List Participants' page of the ANTIVITY platform. At the top, there are navigation links: 'List Participants' (highlighted in red), 'List Competitions', 'List Contests', and 'List Applications'. On the right, there are links for 'staff' and 'Logout'. Below the header, the title 'ค่ายฯลฯ กิจกรรมสำหรับแข่งขัน' is displayed, along with a note 'ต้องรอบ 2 วันเพื่อเตรียมความพร้อมสำหรับการแข่งขันชิงชนะเลิศ' and status indicators: 'Published' and 'ไม่ได้ตรวจสอบ'.

Below the title, there are filters: 'ผู้เข้าร่วม' (selected), 'จำนวน' (0), 'ประเภท' (0), 'สถานะ' (0), and 'บัญชี' (0). A 'Search' button is also present.

The main content area shows a table with columns: 'ลำดับ' (Rank), 'รหัสบัตรประชาชน' (ID Card Number), and 'การดำเนินการ' (Action). One row is visible:

1	6609612160	ตรวจสอบ
---	------------	---------

Below the table, there are three options for action:

- ผู้ได้รับอนุญาต  
ไม่ได้รับอนุญาต
- ยืนยันเข้าร่วมกิจกรรมแล้ว  
ยังไม่ได้รับอนุญาต
- ผู้ที่เข้าร่วมกิจกรรมเรียบร้อยแล้ว  
ยังไม่ได้รับอนุญาต

A 'Delete' button is located at the bottom right of the table.

## 2.14 หน้าจัดการกิจกรรมส่วนผู้เข้าร่วมเมื่อมีคำขอเข้าร่วมกิจกรรมและกดยืนยันแล้ว

The screenshot shows the same 'List Participants' page as the previous one, but the status of the participant has been updated. The row for ID 6609612160 now shows 'อนุมัติ' (Approved) instead of 'ตรวจสอบ' (Checked).

The other parts of the interface remain the same, including the filters, table, and available actions.

2.14 หน้าจัดการกิจกรรมส่วนผู้เข้าร่วมเมื่อผู้เข้าร่วมกดปุ่มยืนยันการเข้าร่วมจะแสดงส่วนการประเมินทักษะและปุ่มกดไม่มาเข้าร่วม

2.15 ส่วนการยืนยันทักษะ ซึ่งสามารถปรับระดับที่ได้และลบทักษะตามที่ได้จริง

## 2.16 หน้าจัดการกิจกรรมส่วนผู้ใช้ร่วมเมื่อมีคำขอเข้าร่วมกิจกรรมและประเมินทักษะให้แล้ว

The screenshot shows the 'User Requests' section of the Antivity application. At the top, there are tabs for 'ผู้เข้าร่วม' (Accepted), 'ผู้ไม่ได้รับอนุมัติ' (Rejected), 'ผู้ที่เข้าร่วมกิจกรรมแล้ว' (User who has participated), and 'ผู้ที่ไม่เข้าร่วมกิจกรรม' (User who has not participated). Below these tabs, there are search and filter options for 'ชื่อ:' (Name:), 'อัลบั้ม:' (Album:), 'รูป:' (Image:), 'เข้าร่วมครั้ง:' (Participation count:), and 'บุรุษ:' (Gender:). A specific request from 'ก้อง' (Kong) is highlighted, showing details such as 'Request ID: 6609612160', 'Status: ผู้ที่เข้าร่วมกิจกรรมแล้ว', and 'ก้อง ภูษิต' (Kong Phusit).

## 2.17 หน้าจัดการหลักสูตรเมื่อนักศึกษาได้หลักสูตรแล้วจะแสดงเปอร์เซ็นต์ความคืบหน้ารวมของหลักสูตร และทักษะที่นักศึกษาในหลักสูตรยังขาด (นับเป็นหน่วย)

The screenshot shows the 'Progress' section of the Antivity application. It displays a grid of student progress information. Each row contains a student's name, the percentage of activities completed, and the number of tasks left. For example, 'ก้อง' (Kong) has completed 61% of his activities and has 2 tasks left.

นักศึกษา	จำนวนกิจกรรมที่เข้าร่วม	จำนวนกิจกรรมที่เหลือ
ก้อง	61%	2

## 2.18 หน้าจัดการนักศึกษาในที่ปรึกษา

รายชื่ออาจารย์  
แบบ สายชม

นักศึกษาของ สมชาย สายชม

รหัส	ชื่อ	ผู้สอน:	อุป
6609612160	ภัทรธน พัฒนา	อนุญาต	ลบ

เพิ่มบันทึกข้อมูลใหม่  
6609612160 - ภัทรธน พัฒนา

## 2.19 ส่วนคุประวัติทักษะ และทักษะของนักศึกษา

รายชื่ออาจารย์  
แบบ สายชม

นักศึกษาของ สมชาย สายชม

รหัส	ชื่อ	ผู้สอน:	อุป
6609612160	ภัทรธน พัฒนา	อนุญาต	ลบ

เพิ่มบันทึกข้อมูลใหม่  
6609612160 - ภัทรธน พัฒนา

กักษะที่มีของ กัณฐ์ ทั้งชัย

ลักษณะ:	ระดับ	วันเดือนปี
การเรียนรู้	2	24 พฤษภาคม 2568
การคิดเชิงวิพากษ์	3	24 พฤษภาคม 2568
การพัฒนาเรียนรู้ตัวเอง	4	24 พฤษภาคม 2568
การท่องเที่ยว	2	24 พฤษภาคม 2568

ประวัติการได้รับกักษะ:

ลักษณะ:	ระดับ	ผู้สอน	หมายเหตุ	วันที่
การท่องเที่ยว	2	ศิรยาภรณ์สักกิจกอร์กินส์สำหรับการเดินทาง	—	24 พฤษภาคม 2568
การพัฒนาเรียนรู้ตัวเอง	4	ศิรยาภรณ์สักกิจกอร์กินส์สำหรับการเดินทาง	—	24 พฤษภาคม 2568
การคิดเชิงวิพากษ์	3	ศิรยาภรณ์สักกิจกอร์กินส์สำหรับการเดินทาง	—	24 พฤษภาคม 2568
การเรียนรู้	2	ศิรยาภรณ์สักกิจกอร์กินส์สำหรับการเดินทาง	—	24 พฤษภาคม 2568

## 2.20 หน้าจัดการกิจกรรมส่วนการประเมิน

The screenshot shows a web-based application interface for managing competition evaluations. At the top, there is a navigation bar with icons for 'กิจกรรมครุภารกิจ' (Curriculum Activities), 'กิจกรรมกีฬา' (Sports Activities), 'กิจกรรมศิลปะและทักษะ' (Arts and Skills Activities), and 'staff'. Below the navigation is a search bar with placeholder text 'ค้นหา...' and a dropdown menu showing 'Published'.

The main content area displays a competition titled 'ค่ายฯลฯลักษณะน้ำหนักการแข่งขัน' (Competitive Programming). It includes a brief description: 'ค่ายฯลฯลักษณะน้ำหนักการแข่งขัน 2 วันเพื่อเตรียมความพร้อมสำหรับการแข่งขันชิงชนะเลิศประเทศไทย (Competitive Programming)'.

Below the title, there are six evaluation criteria with their respective scores and descriptions:

คะแนนมาตรฐาน	คุณธรรม	ความน่าสนใจ	มีเวลา	ความน่าใช้	คะแนนรวม
4.0 คะแนนมาตรฐาน 4	4.0 คุณธรรม 4	4.0 ความน่าสนใจ 4	3.0 มีเวลา 3	5.0 ความน่าใช้ 5	<b>4.0</b> คะแนนรวม 4

At the bottom left, there is a table titled 'บันทึก' (Notes) showing the following data:

คะแนนมาตรฐาน	คุณธรรม
4	4
4	3
5	4

Below the notes table, there are two sections: 'ความต้องการ:' (Requirements) and 'จังหวะและเพื่อนร่วม:' (Scheduling and partners).

On the right side of the page, there is a progress bar labeled 'คะแนนรวม: 4 / 5'.

### 3. นักศึกษา

#### 3.1 หน้าสมัครสมาชิก

ANTIVITY  
สมัครสมาชิกนักศึกษา

รหัสนักศึกษา  
อีเมล รหัสนักศึกษา

ชื่อ-นามสกุล  
อีเมล ชื่อ-นามสกุล

เพศ  
เพศ:

สาขา  
สาขา

ชั้นปี  
ชั้นปี 1

民族  
— กรุณาเลือก —

เบอร์โทรศัพท์  
เบอร์โทรศัพท์

อีเมล  
อีเมล

LINE ID  
อีเมล ID

วันเกิด  
วันเกิด/ปีปัจจุบัน

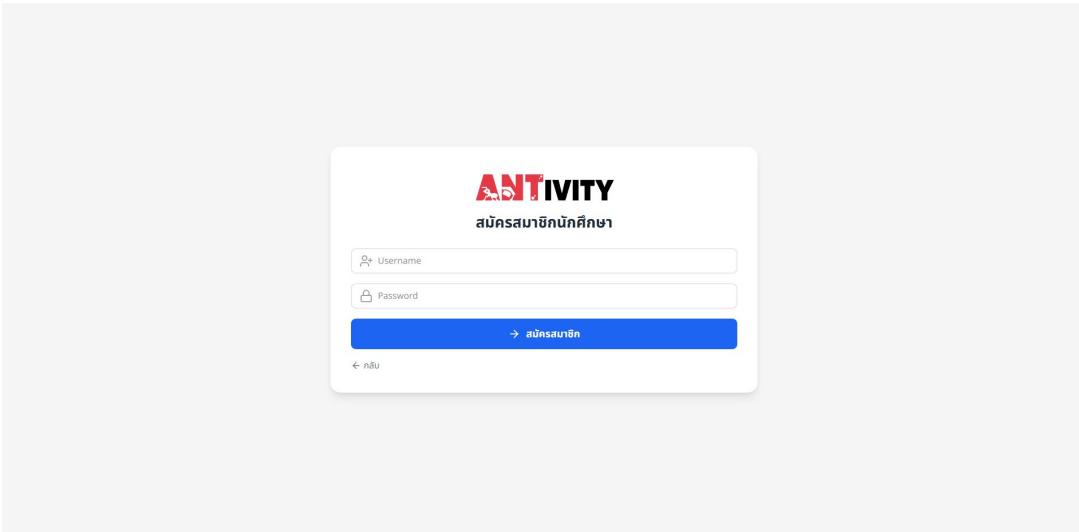
เพศ  
— กรุณาเลือก —

รูปปั๊บ  
↑ เลือกไฟล์ ไม่ได้เลือกไฟล์

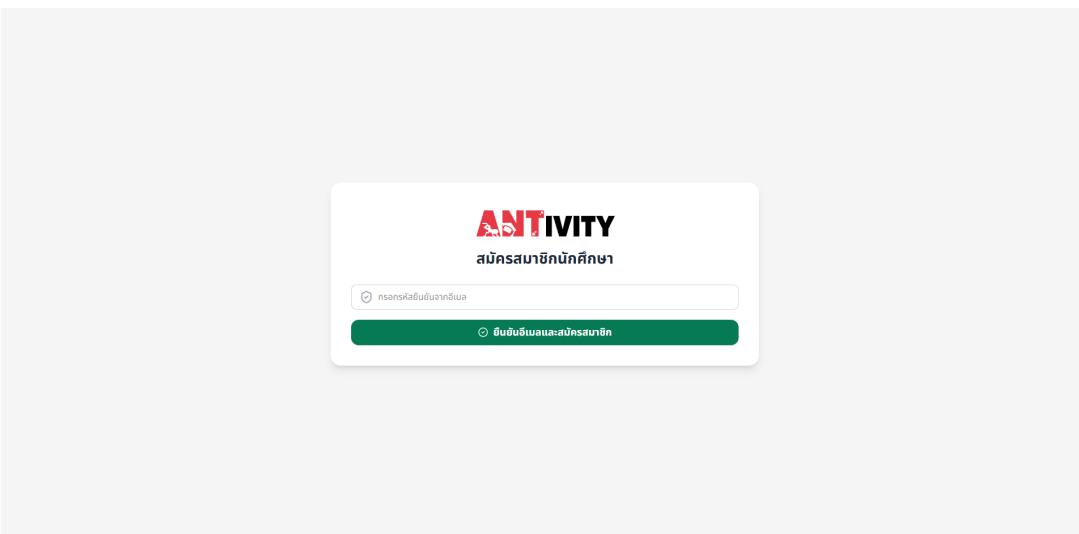
→ ดำเนินการสมัคร (ต่อไป)

มีบัญชีอยู่แล้วใช่ไหม? เข้าสู่ระบบ

### 3.2 ส่วนการกรอก Username และ Password



### 3.3 ส่วนการຢືນຍິ່ນຮັບສິ່ງໄວ້ເອີ້ນເມືລ



### 3.4 หน้าโปรไฟล์บัญชีผู้ใช้

The screenshot shows a user profile for a student named "student". The profile includes basic information like name, ID, and contact details, along with a photo and a "Logout" button.

โปรไฟล์ผู้ใช้

ชื่อผู้ใช้: student  
อีเมล: phattharaphum.kin@dome.tu.ac.th  
ประเภท: student

ข้อมูลบุคคล  
รหัสนักศึกษา: 6609612160  
ชื่อ-นามสกุล: ภัททารพ คงยิ่ง<sup>ก.</sup>  
คณะ: วิทยาศาสตร์และเทคโนโลยี  
สาขาวิชา: วิทยาการคอมพิวเตอร์  
ปี: 2

ออกจากระบบ

### 3.5 หน้าโปรไฟล์นักศึกษา

The screenshot shows a student profile with personal details, academic status, and activity levels.

โปรไฟล์นักศึกษา

ภัททารพ คงยิ่ง (student)  
รหัสนักศึกษา: 6609612160  
ชั้นปี: 2  
อีเมล: phattharaphum.kin@dome.tu.ac.th  
LINE: gamephat  
เพศ: ชาย  
วิทยาการคอมพิวเตอร์  
เพศ: male  
สถานะ: active

ตรวจสอบ  
จำนวนที่ผ่าน: 0% (0/18 หน่วย)

การควบคุม  
สำเร็จแล้ว  
กำลังพัฒนา  
ยังไม่เริ่ม

กิจกรรมที่เคยเข้าร่วม

ยังไม่มีกิจกรรมที่เริ่มที่นี่

### 3.6 หน้าลงทะเบียนเข้าร่วมกิจกรรม

The screenshot shows the ANTIVITY platform's registration interface for the 'Open Hub - II' event. At the top, there are navigation links for 'ไปรษณีย์อิเล็กทรอนิกส์', 'ลงทะเบียนกิจกรรม', 'กิจกรรมของฉัน', and 'student student'. Below the header, the title 'รายการกิจกรรมที่สามารถเข้าร่วมได้' is displayed. The main content area features a banner for 'Open Hub - II' with the tagline 'GAME DESIGN & POWER OF CODING'. The banner includes sections for 'ผู้จัด!!! สำหรับการเรียนต่อศ้าน IT และเทคนิคการเขียน Python + Algorithm ภาษา C++' and 'กิจกรรมที่นักเรียนสามารถเข้าร่วมได้'. Below the banner, there is a summary of requirements: 'ต้องเจ้า: เลือกเลือกหรือเลือกทั้งสอง' and 'แบบฟอร์ม': 'วันเดือนปี พ.ศ. 2568', 'เกณฑ์การประเมิน: Web Development (hard) ดูแล 2', 'Teamwork (soft) ดูแล 2', 'Communication (soft) ดูแล 2', 'Critical Thinking (soft) ดูแล 3'. At the bottom of the page, there are two buttons: 'สมัครสมาชิก' and 'เข้าสู่ระบบ'.

### 3.7 หน้ารายละเอียดกิจกรรมที่สามารถลงทะเบียนได้

The screenshot shows the detailed view of the 'Open Hub - II' event on the ANTIVITY platform. At the top, there are navigation links for 'ไปรษณีย์อิเล็กทรอนิกส์', 'ลงทะเบียนกิจกรรม', 'กิจกรรมของฉัน', and 'student student'. The main content area features a large banner for 'Open Hub - II' with the tagline 'GAME DESIGN & POWER OF CODING'. The banner includes sections for 'ผู้จัด!!! สำหรับการเรียนต่อศ้าน IT และเทคนิคการเขียน Python + Algorithm ภาษา C++' and 'กิจกรรมที่นักเรียนสามารถเข้าร่วมได้'. Below the banner, there are two columns of information: 'ข้อมูลกิจกรรม' and 'กำหนดการ'. The 'ข้อมูลกิจกรรม' column lists: 'ประเภทกิจกรรม: สอน Greedy, Divide & Conquer, DP, Graph ที่มีผลลัพธ์เป็น集合 หรือบวกกับวิชาภาษาต่างๆ เช่นฟังก์ชันเรียกซ้ำ', 'สถานะ: เปิดรับ', 'จำนวนผู้เข้าร่วม: 0/49', 'เวลาเริ่ม: 23 พฤษภาคม 2568', 'เวลาสิ้นสุด: 24 พฤษภาคม 2568', and 'ผู้จัดกิจกรรม: นักเรียน'. The 'กำหนดการ' column lists: 'วันเดือนปี พ.ศ. 2568', 'สถานะ: ยังไม่มี', 'ผู้จัดกิจกรรม: นักเรียน', and 'สถานะ: ยังไม่มี'. At the bottom of the page, there are navigation links for 'กิจกรรมที่นักเรียนสามารถเข้าร่วมได้', 'กิจกรรมที่นักเรียนสามารถเข้าร่วมได้', 'กิจกรรมที่นักเรียนสามารถเข้าร่วมได้', 'กิจกรรมที่นักเรียนสามารถเข้าร่วมได้', and 'กิจกรรมที่นักเรียนสามารถเข้าร่วมได้'.

3.8 หน้ารายละเอียดกิจกรรมเมื่อกดลงที่เป็นเข้าร่วมแล้วรออนุมัติ

The screenshot shows the ANTIIVITY website with the following details:

- Header:** ANTIIVITY logo, navigation links for ฝรั่งฟันธงอัปบัน (Foreigner), ลงทะเบียนกิจกรรม (Register Activity), กิจกรรมของฉัน (My Activities), and student student.
- Section 1 (Top):** A banner for "open IT camp" featuring the text "GAME DESIGN & POWER OF CODING", "พร้อม!!! สำหรับการเรียนรู้ด้าน IT และเทคนิคการเขียน", "Put your brain to work! Algorithm, game design", and "กิจกรรมที่ออกแบบมาเพื่อฝึกอบรมแข่งขัน". It also includes a "LIVE" button and a "สมัคร" (Sign Up) button.
- Section 2 (Left):** A section titled "กิจกรรมกิจกรรม" (Activities) containing:
  - รายละเอียด: กลุ่ม Greedy, Divide & Conquer, DP, Graph ที่มีความซับซ้อนเพิ่มขึ้น หรือเกี่ยวกับภาษาต่อๆ ไป เช่นเชิงปรัชญาและคณิตศาสตร์
  - ลักษณะ: เปิดรับ
  - จำนวนผู้เข้าร่วม: 0/49
  - ลักษณะ: คัดเลือกผู้เข้าร่วมโดยรอบต่อรอบ 2
  - สรุปเม็ดเงิน: 24 พฤษภาคม 2568
  - เม็ดเงินคงเหลือ: 24 พฤษภาคม 2568
- Section 3 (Right):** A section titled "กำหนดการ" (Schedule) containing:
  - วันเดือนปี: 27 พฤษภาคม 2568
  - เปิดรับสมัคร: 24 พฤษภาคม 2568
  - ปิดรับสมัคร: 24 พฤษภาคม 2568 ถึง 26 พฤษภาคม 2568
- Section 4 (Bottom):** A section titled "กิจกรรมที่ควรได้รับ" (Activities to receive) containing:
  - การฝึกอบรมเบื้องต้น (soft) - ระดับ 2
  - การฝึกอบรมเบื้องตื้น (hard) - ระดับ 2
  - การฝึกอบรมทาง (soft) - ระดับ 3
  - การฝึกสอน (soft) - ระดับ 2

### 3.9 หน้ากิจกรรมของฉัน

**ANTIVITY**

ໂປຣໄຟສອນອັນ | **ລອກເບີນກົດກຽບ** | **ຄົກຄອງອັນ** | student  
student

## ກົດກຽບຂອງຈັນ

ກົດທັດ (1)  ຂອບມູນຕີ (1)  ທຶນໄຟເກີນເບີນ (0)  ມືມເກີນເລັວ (0)  ໄນເຫັນຮັນ (0)  ໄນມູນຕີ (0)  ເຫັນຮັນເລັວ (0)  
 ໄນໄດ້ເຫັນຮັນ (0)

### ຄ່າຍເຈະ:ສັກລົກອົກຮັນສໍາເກັນກາ...

27 ພຸດຍຄຸນ 2568

ຄ່າຍເຈະ 2 ນີ້ແມ່ນຄົງຮັນມາດວກເພື່ອກົດກຽບການ  
ແພິ່ນເປົ້າໃນປີປະໂຫຍດ (Competitive...)

ສາມາດກົດກຽບເປົ້າໃນ  
 ສາມາດກົດກຽບໄວ້: ອະວັນນີ້  
 ສາມາດກົດກຽບໄວ້: ຫຼັມໄດ້ເກີນ

**ດຽວນຳເກີນ**

### 3.8 หน้ารายละเอียดกิจกรรมเมื่อได้รับอนุญาตแล้ว และอยู่ในช่วงการยืนยันเข้าร่วมกิจกรรม

The screenshot shows the Antivity platform interface. At the top, there's a navigation bar with icons for login, registration, and student status. The main content area features a banner for an activity titled "open hub IT GAME DESIGN & POWER OF CODING". Below the banner, there's a section for "ผู้อื่น!!! สำหรับการเรียนต่อด้าน IT และเทคนิคการเขียน Python + Algorithm ที่น่าสนใจ" (Others!!! For IT learning and algorithm techniques). A "รับจำนำว่ากัด" (Borrowed) button is visible. The page is divided into two main sections: "ข้อมูลกิจกรรม" (Activity Information) and "กำหนดการ" (Schedule). The "Activity Information" section contains a list of topics: Greedy, Divide & Conquer, DP, Graph (including specific sub-topics like Bellman Ford, Dijkstra, Floyd Warshall), and Dynamic Programming. It also lists submission statistics: 27 ผลงาน (作品) submitted, 24 ผู้เข้าชม (Viewers), and 26 ผู้อ่าน (Readers). The "Schedule" section shows a timeline with tasks: การทำงานเป็นทีม (soft) - รอบที่ 2, การพัฒนาเป็นปกติ (hard) - รอบที่ 2, การติดต่อผู้เชี่ยวชาญ (soft) - รอบที่ 3, การฝึกซ้อม (soft) - รอบที่ 2. At the bottom right, there are buttons for "กติกา" (Rules), "สมัครเข้าร่วม" (Join), and "ออกจากร่วม" (Leave).

### 3.9 หน้ารายละเอียดกิจกรรมเมื่อเมื่อยืนยันเข้าร่วมแล้ว

This screenshot shows the same Antivity platform interface as the previous one, but with a green "ยืนยันแล้ว" (Confirmed) button at the bottom right instead of the "สมัครเข้าร่วm" button. The rest of the content, including the banner, activity details, schedule, and footer buttons, remains identical to the previous screenshot.

3.10 หน้ากิจกรรมของฉันเมื่อเข้าร่วมจะแสดงสถานการณ์ได้รับการประเมินทักษะจากเจ้าหน้าที่ และการทำแบบประเมินกิจกรรม

ANTIVITY
ໂປຣໄຟລ່ອງຈັນ
ສອງເບີນກ່ອກຮຽນ
ກວດສອບຂອບໃຈ
student

## ກົງກຽມຂອງລັບ

ສັນຫຼຸບ (1)
 ຮຳນັກສິນ (0)
 ທຶນໄນເຄືນເຄີນ (0)
 ມີນັກເກົ່າງ (0)
 ໂໍ່ເກົ່າງ (0)
 ນີ້ແມ່ນເສີ (0)
 ເຕັກວັນເຕັກ (1)

ໂໍ່ໄດ້ເຕັກວັນ (0)

ຄ່າຍເຈາະເລັກວັດທີຮັບສໍາເລັກກາ...

□ 27 ພຸດຍາດ 2568

ຄ່າຍເຈາະ 2 ວິທີເຕັກວັດທີຮັບສໍາເລັກກາ  
ແພື່ນເປົ້າປົ້ນປົກປຽບ

- ສາມາດກົດກົດ ເຊິ່ງເລີ້ມ
- ສາມາດກົດກົດຢ່າງວັນ ເຊິ່ງວັນເລີ້ມ
- ສາມາດກົດກົດຢ່າງວັນ ອັນເປັດເລີ້ມ
- ສາມາດກົດກົດຢ່າງວັນ ປະກິບກິບ: ປະເມັນເລີ້ມ
- ແບບປະເມັນກ່ອກຮຽນ: ສ້າງໄປເຕີສຳ

ດູວຍການເລີ້ມ

กิจกรรม 3.11 หน้ารายละเอียดกิจกรรมเมื่อกิจกรรมเสร็จสิ้น ได้รับการประเมินทักษะแล้ว และยังไม่ได้ประเมิน

ANTIVITY

ໂປຣໄຟຈົກຂອນ | ລວກປະເປີກອດກຽມ | ກອກຮົມຈົກຂອນ | student student

# openITup

GAME DESIGN & POWER OF CODING

ພ່ອມ!!! ສໍາເລັກທັງເຮົາໃຫຍ້ຄ້ານ IT ແລະ ເຫດຜົນການເປັນ

Put your Algorithm into Action

ຄ່າຍອດກົດກົດສ່ວນເຫັນການເປັນ

ຄ່າຍອດ 2 ໂບນາໂດຍຮັບການເປັນການປະກວດປະກວດ (Competitive Programming)

ສໍານັກປະກວດ

LIVE

ຮັບຈຳນວນຈຳກັດ

OpenITup

### ໜ້າມຸກຕອກຮົມ

- ຮາມເນື້ອ: ລວມ Greedy, Divide & Conquer, DP, Graph ໄກສອນເປັນເຂົ້າຂຶ້ນ ຢ່ວມເຖິງມີວິທີການຈາກຊູ້ເໝັ້ນເປັນເປົ້າປະກາດ
- ລາມາດ: [ເຊີ່ມຕົ້ນ](#)
  - ຮັບຈຳນວນຈຳກັດ: 1/49
  - ສຳຄັນ: ສ່ວນປົງດີກົດກອບພົວເວລີ 2
  - ສຳຄັນ: 24 ພຸດຍາກນ 2568
  - ລົບຕົວຕ່າງໆ: 24 ພຸດຍາກນ 2568

### ກໍານົດກາ

- ວັນອີກຕົກຮົມ: 27 ພຸດຍາກນ 2568
- ປີເປັນເປົ້າທີ: 24 ພຸດຍາກນ 2568
- ດີເປັນເປົ້າທີ: 24 ພຸດຍາກນ 2568 ໂດຍ 26 ພຸດຍາກນ 2568

### 3.12 หน้าประเมินกิจกรรม

**ANTIVITY**

ประเมินกิจกรรม | ลงทะเบียนกิจกรรม | กิจกรรมของฉัน | student

**แบบประเมินกิจกรรม**  
☆ ให้คะแนน (1-5)

การอธิบายและสอนที่  
 1  2  3  4  5

ผู้บรรยาย / ผู้ชี้ดู  
 1  2  3  4  5

ความบันเทิง  
 1  2  3  4  5

เมื่อค่า

การนำไปใช้ประโยชน์  
 1  2  3  4  5

ภาพรวมกิจกรรม

ความต้องการที่  
ความต้องการที่นำไปใช้ประโยชน์  
ความต้องการที่นำไปใช้ประโยชน์

ฉันไม่เปิดเผยหรือ

**ANTIVITY**

ประเมินกิจกรรม | ลงทะเบียนกิจกรรม | กิจกรรมของฉัน | student

**แบบประเมินกิจกรรม**  
☆ ให้คะแนน (1-5)

การอธิบายและสอนที่  
 1  2  3  4  5

ผู้บรรยาย / ผู้ชี้ดู  
 1  2  3  4  5

ความบันเทิง  
 1  2  3  4  5

เมื่อค่า

การนำไปใช้ประโยชน์  
 1  2  3  4  5

ภาพรวมกิจกรรม

ความต้องการที่  
ความต้องการที่นำไปใช้ประโยชน์  
ความต้องการที่นำไปใช้ประโยชน์

ฉันไม่เปิดเผยหรือ

### 3.11 หน้ารายละเอียดกิจกรรมเมื่อกิจกรรมเสร็จสิ้น ได้รับการประเมินทักษะแล้ว และประเมินทักษะแล้ว

The screenshot shows the ANTIIVITY platform interface. At the top, there are navigation links: 'ไม่ใช่อีเมล' (Not using email), 'ลงทะเบียนผู้สอน' (Register teacher), 'กิจกรรมของฉัน' (My activities), and 'student student'. Below this is a banner for 'Open Hub - II' with the text 'GAME DESIGN & POWER OF CODING' and 'ห้าม!!! สำหรับการเรียนต่อข้าน IT และเทคนิคการเขียน Putbox + Algorithm ตาม'.

**ข้อมูลกิจกรรม:**

- รายละเอียด: แบบ Greedy, Divide & Conquer, DP, Graph คือการคิดแบบที่ซึ่งเรา หรือที่มีวิธีการของผู้เข้าแข่งขันเป็นไปได้
- สถานะ: ยังไม่ได้รับ: 1/49
- จำนวนผู้เข้าร่วม: 24
- จำนวนผู้ชนะ: 2
- จำนวนผู้เข้าร่วม: 24 พฤษภาคม 2568
- จำนวนผู้ชนะ: 24 พฤษภาคม 2568 กว่า 26 พฤษภาคม 2568

**กิจกรรมที่เข้าร่วม:**

- การเขียนโปรแกรม (soft) - ระดับ 2
- การเขียนโปรแกรม (hard) - ระดับ 2
- การเขียนภาษา Python (soft) - ระดับ 3
- การเขียน (soft) - ระดับ 2

On the right side, there is a sidebar with the title 'ค่ายเจ้าเล็กอัลกอริทึมสำหรับการ...' and a list of filters:
 

- วันที่: 27 พฤษภาคม 2568
- ค่ายอบรม 2 วันเพื่อเตรียมความพร้อมสำหรับการแข่งขันโปรแกรม (Competitive Programming)
- สถานะ: กิจกรรม: เริ่มแล้ว
- สถานะ: การเขียนข้อสอบ: ยังเขียนแล้ว
- สถานะ: การเขียนข้อสอบ: ประมวลผลแล้ว
- แบบประเมินกิจกรรม: ส่งแล้ว

### 3.12 หน้าโปรไฟล์นักศึกษาหลังเข้าร่วมกิจกรรมข้างต้น

The screenshot shows the ANTIIVITY platform interface. At the top, there are navigation links: 'ไม่ใช่อีเมล' (Not using email), 'ลงทะเบียนผู้สอน' (Register teacher), 'กิจกรรมของฉัน' (My activities), and 'student student'.

**โปรไฟล์นักศึกษา:**

ชื่อ: พัฒน์ภูมิ ถึงขัย  
รหัส: 6609612160  
อีเมล: phatthraphum.kin@dome.tu.ac.th  
LINE: gamephat  
เพศ: male  
สถานะ: active  
ภาษาที่พูด: English

ความคืบหน้า: 61% (11/18 หน่วย)

**กิจกรรมที่เคยเข้าร่วม:**

Open Hub - II

ค่ายเจ้าเล็กอัลกอริทึมสำหรับการแข่งขัน

On the right side, there is a sidebar with the title 'กิจกรรมที่เข้าร่วม' and a list of filters:
 

- กิจกรรมที่เข้าร่วม: 4/4
- จำนวนผู้เข้าร่วม: 2/4
- จำนวนผู้ชนะ: 3/4
- จำนวนผู้เข้าร่วม: 2/4
- จำนวนผู้ชนะ: 0/2

## 4. อาจารย์ที่ปรึกษา

### 4.1 หน้าสมัครสมาชิกอาจารย์

The image displays three vertically stacked screenshots of a web-based application for teacher registration. The top screenshot shows a full-page registration form with various input fields and a large blue 'Submit' button. The middle screenshot shows a simplified version of the same form, likely a step in the process. The bottom screenshot shows a confirmation or success page with a green 'OK' button.

**Screenshot 1: Full Registration Form**

This screenshot shows a comprehensive registration form for teachers. It includes fields for:

- ชื่อ-นามสกุล (Name)
- เพศ (Gender)
- โทรศัพท์ (Phone)
- เลือกตามนี้ (Select one)
- เบอร์โทรศัพท์ (Phone number)
- อีเมล (Email)
- ไฟล์แนบ (Attachment) with a 'Choose File' button

A large blue button at the bottom right says "→ ดำเนินการต่อไป (Next)". Below it is a link: "มีบัญชีอยู่แล้วให้เข้า? เข้าสู่ระบบ".

**Screenshot 2: Simplified Registration Form**

This screenshot shows a simplified version of the registration form, likely a step in the process. It includes fields for:

- Username
- Password

A large blue button at the bottom right says "→ ดำเนินการต่อไป (Next)". Below it is a link: "← กลับ".

**Screenshot 3: Confirmation Page**

This screenshot shows a confirmation or success page. It includes a checkbox for accepting terms and conditions and a large green button at the bottom right that says "ยืนยันอีเมลและบัญชีของฉัน (Confirm my email and account)".

## 4.2 หน้าโปรไฟล์บัญชีผู้ใช้อาจารย์

ANTIVITY

ໂປຣໄຟລ໌ອອນໄລນ໌ | ສະບັບອົກສາ | professor

### ໂປຣໄຟລ໌ຜູ້ໃຊ້

ຊື່ຜູ້ໃຊ້: pf  
ອີເມວ: gameponophaof@gmail.com  
ສາບາດ: professor

ຂໍອະຫຍາດ  
ທີ່ຢູ່-ບານຄຸກ: ສນອຍ ລາຍນ

ອອກຈາກໂປຣ

## 4.3 หน้าโปรไฟล์อาจารย์

ANTIVITY

ໂປຣໄຟລ໌ອອນໄລນ໌ | ສະບັບອົກສາ | professor

ຜູ້ບໍ່ຢູ່ຄາສຕຣາຈາຍ ດຣ. ສນອຍ ສາຍນ

ອີເມວ: gameponophaof@gmail.com	ເອົາເວີຣີ: 0888888888
ການອົງ: ວິທາການຄົນພິຈາລະນາ	ຄະນະ: ວິທາການລະບົບ
ສຳເນົານັ້ນ: ຜູ້ອ່ານຄະດວງລາຍລະອຽດ	ສານມະນຸຍົງ: ໃຈນາ
ລົງທຶນເລີ້ມ: 5/24/2025, 9:50:04 PM	ເກີ່ອຫຼືຂູ້ຮູ່: 5/24/2025, 9:50:04 PM

#### 4.4 หน้าภาพรวมนักศึกษาในความดูแล

The screenshot shows a user profile for 'กักรถยนต์ กิ่งชัย' (Kinavit). The top navigation bar includes the ANTIVITY logo, a user icon, and the text 'โปรไฟล์ของฉัน' (My Profile) and 'ผู้สอน' (Teacher). Below the navigation is a progress bar indicating 61% completion. A blue button labeled 'ไปที่โปรไฟล์' (Go to Profile) is visible. The main content area displays the student's name, ID (6609612160), and gender (ชาย). It also lists their LINE ID (@phattharaphum.kin@dome.tu.ac.th) and phone number (0999632302). To the right, there are sections for 'กิจกรรมที่เคยเข้าร่วม' (Activities participated in) and 'กิจกรรมที่กำลังดำเนินการ' (Activities currently ongoing). The 'กิจกรรมที่เคยเข้าร่วม' section includes 'Open Hub - II' with a yellow banner for 'LIVE'. The 'กิจกรรมที่กำลังดำเนินการ' section shows four items: 'กิจกรรมนักศึกษา' (Student Activity), 'กิจกรรมนักเรียน' (Student Activity), 'กิจกรรมนักศึกษา' (Student Activity), and 'กิจกรรมนักเรียน' (Student Activity).

#### 4.5 หน้าดูໂປຣໄຟລ໌ນັກສຶກສາ

The screenshot shows a teacher profile for 'กักรถยนต์ กิ่งชัย' (Kinavit). The top navigation bar includes the ANTIVITY logo, a user icon, and the text 'โปรไฟล์ของฉัน' (My Profile) and 'ผู้สอน' (Teacher). Below the navigation is a progress bar indicating 61% completion. The main content area displays the teacher's name, ID (6609612160), and gender (ชาย). It also lists their LINE ID (@phattharaphum.kin@dome.tu.ac.th) and phone number (0999632302). To the right, there are sections for 'กิจกรรมที่เคยเข้าร่วม' (Activities participated in) and 'กิจกรรมที่กำลังดำเนินการ' (Activities currently ongoing). The 'กิจกรรมที่เคยเข้าร่วม' section includes 'Open Hub - II' with a yellow banner for 'LIVE'. The 'กิจกรรมที่กำลังดำเนินการ' section shows four items: 'กิจกรรมนักศึกษา' (Student Activity), 'กิจกรรมนักเรียน' (Student Activity), 'กิจกรรมนักศึกษา' (Student Activity), and 'กิจกรรมนักเรียน' (Student Activity).