

แบบรายงานความก้าวหน้าในการทำโครงการ

คะแนน (เต็ม 10)

โครงการ เว็บแอปพลิเคชันสำหรับระบบติดตามสถานะนักศึกษาระดับบัณฑิตศึกษา

รายงานครั้งที่ 2

(ลายมือชื่อ
อ.ที่ปรึกษา)

1. ภาพรวม (Summary)

งานด้านเทคนิค

งาน (Task)	นายนิติภูมิ จันทร์อำ	นายภัทรพล พลกุล	ร้อยละ ความสำเร็จตาม แผนงาน
ศึกษาการใช้ API ใน Postman(ร้อยละ 100)	3 ชั่วโมง	3 ชั่วโมง	100
ทดลองการใช้งาน API ใน Postman(ร้อยละ 100)	3 ชั่วโมง	3 ชั่วโมง	100
ออกแบบหน้าตาผู้ใช้งาน UX/UI ของเว็บแอปพลิเคชันสำหรับระบบติดตามสถานะนักศึกษา ระดับบัณฑิตศึกษา (ร้อยละ 100)	3 ชั่วโมง	3 ชั่วโมง	100
แก้ไขหน้าตาผู้ใช้งาน UX/UI ของเว็บแอปพลิเคชันสำหรับระบบติดตามสถานะนักศึกษา ระดับบัณฑิตศึกษา (ร้อยละ 100)	3 ชั่วโมง	3 ชั่วโมง	100

งานทั่วไป

งาน (Task)	นายนิติภูมิ จันทร์อำ	นายภัทรพล พลกุล	ร้อยละ ความสำเร็จตาม แผนงาน
ศึกษาวิธีการเพิ่มและลบข้อมูลและอัปเดตข้อมูลผ่านหน้าเว็บ (ร้อยละ 100)	3 ชั่วโมง	3 ชั่วโมง	100
ศึกษาภาษา Python ในการเรียกข้อมูลจากฐานข้อมูล (ร้อยละ 100)	3 ชั่วโมง	3 ชั่วโมง	100

การศึกษาการออกแบบ UX/UI (ร้อยละ 100)	3 ชั่วโมง	3 ชั่วโมง	100
--------------------------------------	-----------	-----------	-----

2. ผลสัมฤทธิ์สำคัญในสัปดาห์ที่ผ่านมา (Highlights)

- 1. เพิ่มและลบข้อมูลและอัปเดตข้อมูลผ่านหน้าเว็บ
- 2. เพิ่มข้อมูลและเรียกข้อมูลโดยใช้ภาษา python
- 3. ออกแบบหน้าต่างผู้ใช้งาน UX/UI ของเว็บแอปพลิเคชันสำหรับระบบติดตามสถานะนักศึกษาในระดับบัณฑิตศึกษา
- 4. แก้ไขหน้าต่างผู้ใช้งาน UX/UI ของเว็บแอปพลิเคชันสำหรับระบบติดตามสถานะนักศึกษาในระดับบัณฑิตศึกษาให้สมบูรณ์

3. รายละเอียดในการปฏิบัติงาน (Details)

3.1. รายละเอียดของทฤษฎีที่เกี่ยวข้องกับการพัฒนาเว็บแอปพลิเคชัน

- ทฤษฎีการพัฒนาเว็บแอปพลิเคชัน (Web Application Development) คือ เว็บแอปพลิเคชันที่พัฒนาสำหรับระบบติดตามนักศึกษาในระดับบัณฑิตศึกษาเป็นซอฟต์แวร์ที่สามารถทำงานบนเว็บเบราว์เซอร์ ทำให้นักศึกษาและเจ้าหน้าที่สามารถเข้าถึงและใช้งานระบบได้ผ่านอินเทอร์เน็ต ไม่ว่าจะอยู่ที่ใดก็ตาม ซึ่งช่วยลดความจำเป็นในการเดินทางมายังคณะเพื่อดำเนินการเอกสาร ระบบนี้มีการออกแบบให้ผู้ใช้สามารถกรอกแบบฟอร์มและติดตามสถานะของเอกสารได้อย่างสะดวกและรวดเร็ว

ในส่วนของ Front-End Development ซึ่งเป็นส่วนที่นักศึกษาหรือเจ้าหน้าที่จะได้ตอบกับระบบผ่านหน้าเว็บ UI ควรมีการใช้เทคโนโลยี HTML, CSS และ JavaScript เพื่อสร้างประสบการณ์ผู้ใช้ (User Experience, UX) ที่ดี โดยหน้าเว็บจะต้องใช้งานง่ายและเป็นมิตรต่อผู้ใช้ ในขณะที่ Back-End Development ใช้เฟรมเวิร์คอย่าง Django ที่ทำงานบนภาษา Python สำหรับจัดการข้อมูลและการทำงานทางธุรกิจ เช่น การประมวลผลข้อมูลการยื่นเอกสาร การอัปเดตสถานะ และการจัดการข้อมูลผู้ใช้

- เฟรมเวิร์ค Django และการพัฒนาระบบ Back-End คือ การพัฒนา Back-End ของระบบนี้ใช้ Django Framework ซึ่งเป็นเครื่องมือที่ทรงพลังสำหรับการพัฒนาเว็บแอปพลิเคชันในภาษา Python โดย Django มีระบบจัดการผู้ใช้ (User Authentication) ที่สามารถนำไปใช้กับการยืนยันตัวตนผ่านอีเมลล์ของมหาวิทยาลัยได้ เพื่อให้สิทธิ์การเข้าถึงและจัดการข้อมูลระหว่างนักศึกษา เจ้าหน้าที่ และอาจารย์แตกต่างกันไป ระบบนี้ใช้

โครงสร้าง Model-View-Template (MVT) ที่ช่วยจัดการข้อมูลในฐานข้อมูลและแสดงผลข้อมูลบนหน้าเว็บ การแยกตรรกะการทำงานออกเป็นส่วนๆ ทำให้การพัฒนาง่ายต่อการดูแลรักษา

ในกรณีของระบบติดตามเอกสาร การใช้ Django ช่วยให้การจัดการเอกสารทั้งหมดเป็นระบบดิจิทัล สามารถบันทึกและดึงข้อมูลจากฐานข้อมูลได้โดยตรง ทำให้เจ้าหน้าที่ไม่ต้องเสียเวลาบันทึกข้อมูลลงในไฟล์ Excel นอกจากนี้ Django ยังรองรับการเชื่อมต่อกับ API เพื่อรับและส่งข้อมูลระหว่างเว็บแอปพลิเคชันและโมบายแอปพลิเคชัน

- ทฤษฎีการออกแบบส่วนติดต่อผู้ใช้ (User Interface Design) และประสบการณ์ผู้ใช้ (User Experience) คือ ระบบนี้จำเป็นต้องมีการออกแบบส่วนติดต่อผู้ใช้ (UI) ที่เรียบง่ายและสะดวกในการใช้งาน เนื่องจากผู้ใช้เป็นทั้งนักศึกษา เจ้าหน้าที่ และอาจารย์ การออกแบบ Responsive Web Design เป็นสิ่งสำคัญเพื่อให้ระบบสามารถปรับการแสดงผลให้เหมาะสมกับหน้าจอขนาดต่างๆ ทั้งคอมพิวเตอร์และอุปกรณ์พกพา เทคโนโลยี HTML และ CSS ถูกใช้ในการสร้างโครงสร้างและการจัดวางองค์ประกอบหน้าเว็บ ขณะที่ JavaScript ใช้ในการเพิ่มความสามารถในการโต้ตอบและทำให้การแสดงผลข้อมูลในระบบเป็นแบบไดนามิก

- ฐานข้อมูลและการจัดการข้อมูล (Database Management) คือ เว็บแอปพลิเคชันนี้มีความต้องการในการจัดการข้อมูลเอกสารจำนวนมากจากนักศึกษาระดับบัณฑิตศึกษา เช่น ข้อมูลการยื่นคำร้องขออนุมัติหัวข้อวิทยานิพนธ์ และข้อมูลสถานะการอนุมัติ การจัดการข้อมูลจึงต้องใช้ Relational Database Management System (RDBMS) อย่าง MySQL หรือ PostgreSQL ซึ่งจะช่วยให้การจัดการข้อมูลมีความเป็นระบบ สามารถค้นหาและอัปเดตข้อมูลได้อย่างรวดเร็ว ระบบนี้ออกแบบให้มีการจัดเก็บข้อมูลแบบดิจิทัล ทำให้เจ้าหน้าที่สามารถเข้าถึงและจัดการข้อมูลได้โดยไม่ต้องจัดการเอกสารในรูปแบบกระดาษ ซึ่งเป็นการลดภาระงานและข้อผิดพลาดที่เกิดจากการจัดเก็บข้อมูลแบบเดิม

- การเชื่อมต่อผ่าน API (Application Programming Interface) คือ ในระบบติดตามเอกสารนี้มีการพัฒนา API เพื่อเชื่อมต่อกับระบบภายนอก เช่น โมบายแอปพลิเคชันที่นักศึกษาจะใช้ในการตรวจสอบสถานะของเอกสาร การออกแบบ API ที่ดีช่วยให้ระบบสามารถสื่อสารและส่งข้อมูลกันได้อย่างมีประสิทธิภาพ โดย API จะถูกใช้ในการยืนยันตัวตน การดึงข้อมูลสถานะเอกสาร การอัปเดตสถานะ และการแจ้งเตือนเหตุการณ์สำคัญผ่านอีเมล นอกจากนี้ API ยังมีการรักษาความปลอดภัยข้อมูล โดยใช้การเข้ารหัสข้อมูลระหว่างการรับส่ง และการตรวจสอบสิทธิ์ผู้ใช้ในการเข้าถึงข้อมูล

3.2. รายละเอียด เนื้อหาทางทฤษฎี หลักการออกแบบ web framework Django โครงสร้างของระบบที่ได้จากการออกแบบ

- หลักการออกแบบเว็บแอปพลิเคชันด้วย Django เป็นหนึ่งในเว็บเฟรมเวิร์คที่พัฒนาด้วยภาษา Python มีจุดเด่นในการเป็น "Batteries Included" นั้นหมายความว่า Django มีฟังก์ชันและเครื่องมือที่จำเป็นสำหรับการพัฒนาเว็บแอปพลิเคชันแบบครบครัน เช่น ระบบจัดการผู้ใช้ ระบบยืนยันตัวตน ระบบฐานข้อมูล ระบบการจัดการไฟล์ และระบบการแสดงผลข้อมูล Django ช่วยลดเวลาในการพัฒนาแอปพลิเคชันด้วยการจัดเตรียมโครงสร้างพื้นฐานและส่วนประกอบสำคัญไว้ล่วงหน้า

- หลักการออกแบบที่ใช้ใน Django คือการแยกตรรกะการทำงานออกเป็นส่วน ๆ โดยแบ่งเป็น 3 ส่วนหลัก ประกอบด้วย

- Model - เป็นส่วนที่จัดการข้อมูลในฐานข้อมูล ซึ่งจะรับผิดชอบในการสร้าง แก้ไข ลบ และเรียกข้อมูลจากฐานข้อมูล Model ใน Django มีโครงสร้างที่สอดคล้องกับตารางข้อมูลในฐานข้อมูล (Relational Database) เช่น MySQL หรือ PostgreSQL ในระบบติดตามนักศึกษาชั้น ข้อมูลต่าง ๆ เช่น ข้อมูลนักศึกษา ข้อมูลเอกสาร และสถานะของเอกสาร จะถูกจัดเก็บในฐานข้อมูลผ่าน Model
- View - เป็นส่วนที่รับข้อมูลจาก Model และจัดการกับตรรกะทางธุรกิจ (Business Logic) ก่อนส่งข้อมูลไปยังผู้ใช้ผ่านส่วนติดต่อผู้ใช้ (User Interface) ในระบบนี้ View จะรับผิดชอบการดึงข้อมูลเอกสาร สถานะการอนุมัติ การยื่นเอกสาร และการอัปเดตสถานะต่าง ๆ ของเอกสารที่ส่งมาจากนักศึกษาและเจ้าหน้าที่
- Template - เป็นส่วนที่จัดการการแสดงผลข้อมูลหรือ UI (User Interface) ให้กับผู้ใช้ ซึ่งจะใช้เทคโนโลยี HTML, CSS และ JavaScript ในการออกแบบให้การแสดงผลสวยงาม ใช้งานง่าย ในระบบนี้หน้าแดชบอร์ดจะแสดงสถานะเอกสารต่าง ๆ เช่น วิทยานิพนธ์และคำร้องที่นักศึกษายื่นไป

- โครงสร้างของระบบตามการออกแบบ คือ การออกแบบโครงสร้างระบบติดตามนักศึกษาระดับบัณฑิตศึกษาผ่านเว็บแอปพลิเคชันนี้ มีโครงสร้างที่สอดคล้องกับสถาปัตยกรรม Model-View-Template (MVT) ซึ่งเป็นหลักการออกแบบระบบใน Django โดยสามารถแบ่งเป็น 3 ส่วนหลัก ดังนี้

- Frontend (User Interface): นักศึกษา เจ้าหน้าที่ และอาจารย์ สามารถโต้ตอบกับระบบผ่านหน้า UI ที่ออกแบบมาให้ใช้งานง่ายและเป็นมิตรกับผู้ใช้ (User-Friendly) เทคโนโลยีที่ใช้ในส่วนนี้ประกอบไปด้วย HTML, CSS และ JavaScript โดยที่หน้าแดชบอร์ดหลักจะแสดงข้อมูลสถานะเอกสารการอนุมัติของนักศึกษา ระบบจะทำการแจ้งเตือนเมื่อนักศึกษามีการเปลี่ยนแปลงสถานะเอกสาร หรือต้องมีการอัปเดตข้อมูล
- Backend (Server Side Logic): ส่วนนี้ใช้ Django ในการจัดการตรรกะทางธุรกิจ การประมวลผลข้อมูลจากฐานข้อมูล และการจัดการคำร้องขอจากผู้ใช้ เช่น การอัปเดตสถานะเอกสาร การจัดเก็บและดึงข้อมูลจากฐานข้อมูล ระบบการยืนยันตัวตนผ่านอีเมลมหาวิทยาลัยและการให้สิทธิ์การเข้าถึงข้อมูลต่าง ๆ เช่น นักศึกษา เจ้าหน้าที่ และอาจารย์ ระบบ Backend จะส่งข้อมูลไปยัง Frontend เพื่อนำไปแสดงผลบนหน้าเว็บ
- Database Management: ในระบบนี้ใช้ฐานข้อมูลแบบ Relational Database ซึ่งเหมาะสำหรับการจัดเก็บข้อมูลที่มีโครงสร้างเป็นตาราง เช่น ข้อมูลนักศึกษา ข้อมูลเอกสาร และสถานะต่าง ๆ ของเอกสาร โดยฐานข้อมูลนี้จะเชื่อมต่อกับ Django ผ่าน ORM (Object-Relational Mapping) ที่ช่วยให้การจัดการข้อมูลในฐานข้อมูลมีความสะดวกและปลอดภัยมากขึ้น Django ORM จะช่วยในการสร้างตารางข้อมูลจาก Model ที่ออกแบบไว้โดยไม่ต้องเขียน SQL เอง

- ความปลอดภัยของระบบ (Security) การพัฒนาระบบติดตามนักศึกษานี้ใช้ ระบบการยืนยันตัวตน (Authentication) ผ่านอีเมลของมหาวิทยาลัย ซึ่งเป็นระบบที่ปลอดภัยในการตรวจสอบสิทธิ์ของผู้ใช้ รวมถึงการจัดการสิทธิ์การเข้าถึงข้อมูล (Authorization) โดย Django มีระบบจัดการผู้ใช้ (User Management System) ในตัว ทำให้สามารถควบคุมการเข้าถึงข้อมูลของผู้ใช้ในแต่ละระดับ เช่น นักศึกษา อาจารย์ และเจ้าหน้าที่ ได้อย่างมีประสิทธิภาพ

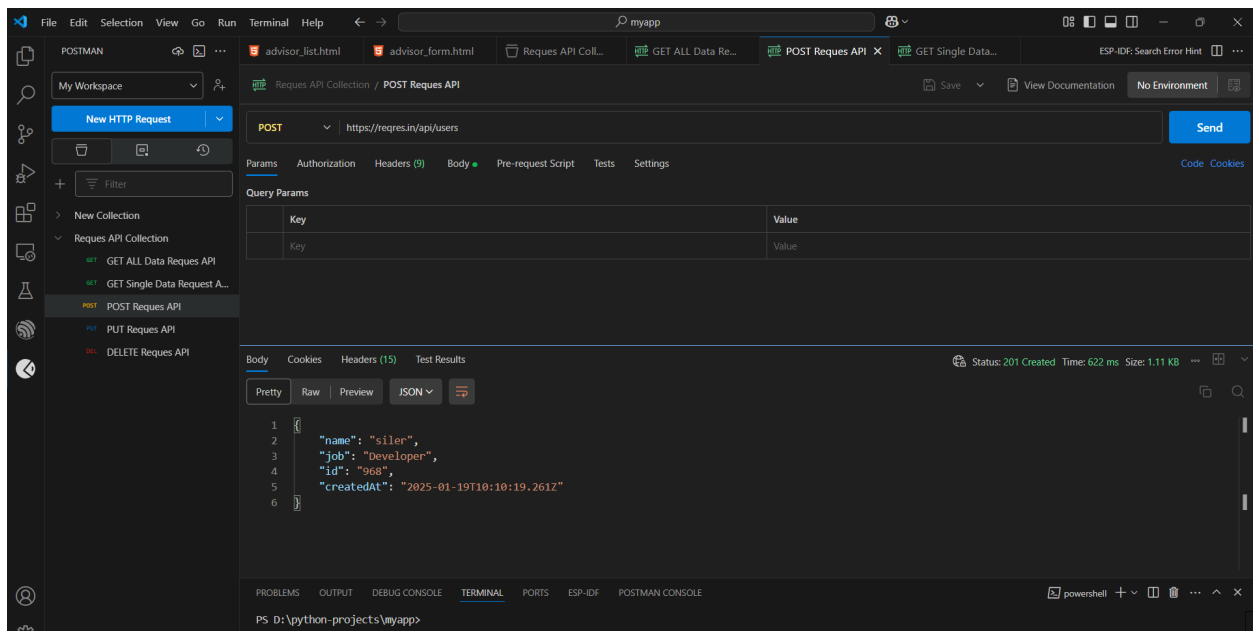
- การเชื่อมต่อกับ API (Application Programming Interface) ระบบติดตามนักศึกษานี้ยังมีการออกแบบ API เพื่อใช้ในการเชื่อมต่อระหว่างเว็บแอปพลิเคชันกับโมบายแอปพลิเคชันของนักศึกษา ซึ่ง API จะทำหน้าที่เป็นสะพานกลางในการรับส่งข้อมูล เช่น การยืนยันตัวตน การอัปเดตสถานะเอกสาร หรือการแจ้งเตือนข้อมูลสำคัญ โดยการใช้ Django Rest Framework (DRF) ช่วยในการพัฒนา API ให้มีความยืดหยุ่นและประสิทธิภาพสูง

API ที่ถูกพัฒนาขึ้นจะช่วยให้โมบายแอปพลิเคชันสามารถดึงข้อมูลสถานะเอกสารของนักศึกษาหรือข้อมูลสำคัญอื่น ๆ ได้ทันทีผ่านอินเทอร์เน็ตที่ออกแบบไว้ ซึ่งช่วยเพิ่มความสะดวกในการติดตามเอกสารแบบเรียลไทม์ผ่านอุปกรณ์พกพา

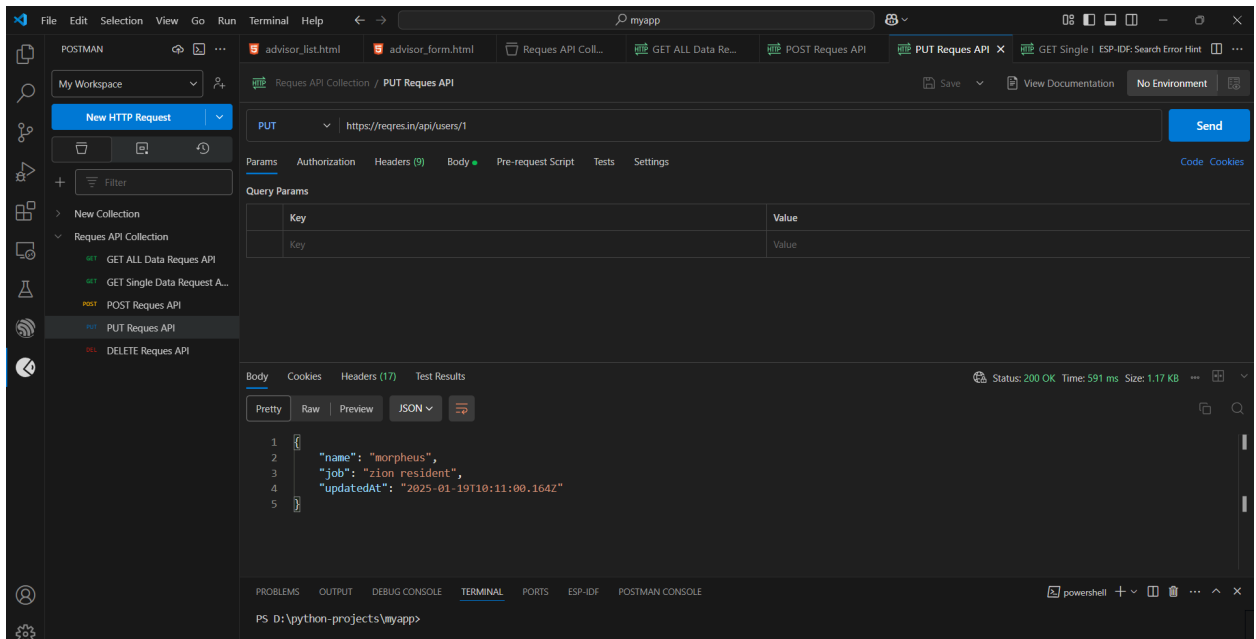
3.3. รายละเอียดในการปฏิบัติงาน (Details)

- ส่วนของข้อมูลทั้งหมด

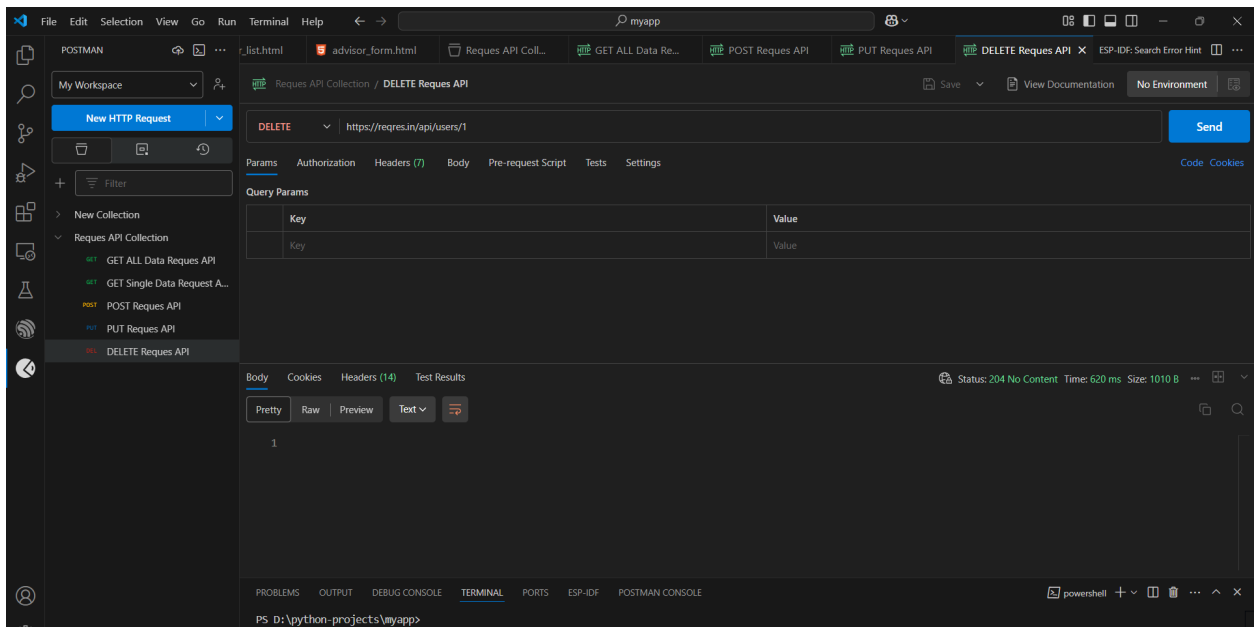
- ส่วนเพิ่มข้อมูล



- ส่วนของแก้ไขข้อมูล

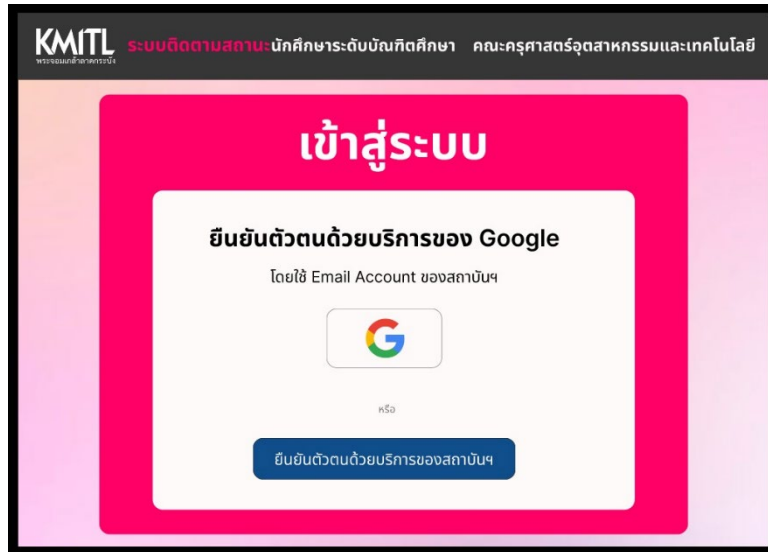


- ส่วนของการลบข้อมูล



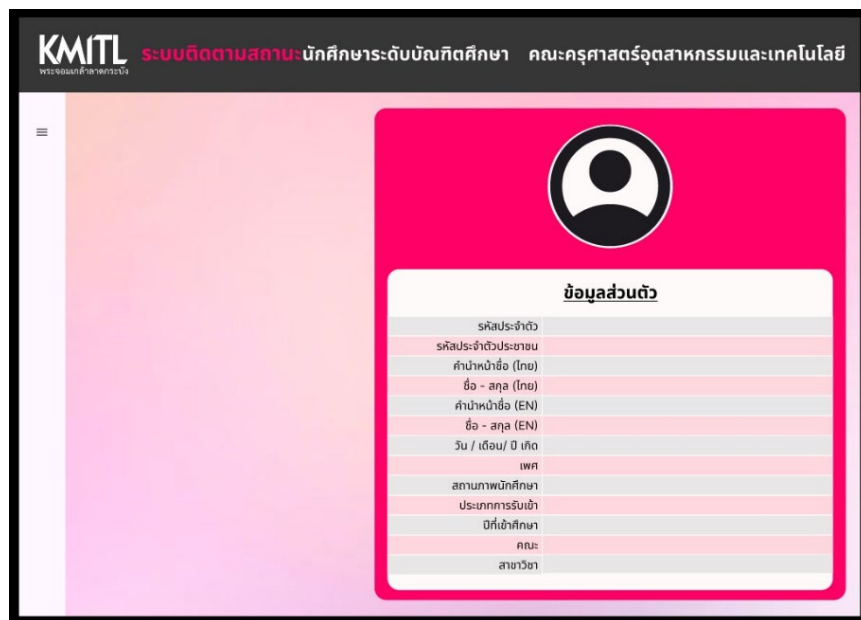
3.4. การออกแบบ UX/UI ของหน้าเว็บไซต์ และหน้า Report ที่ให้ user ใช้งาน

-หน้าต่อการล็อกอิน

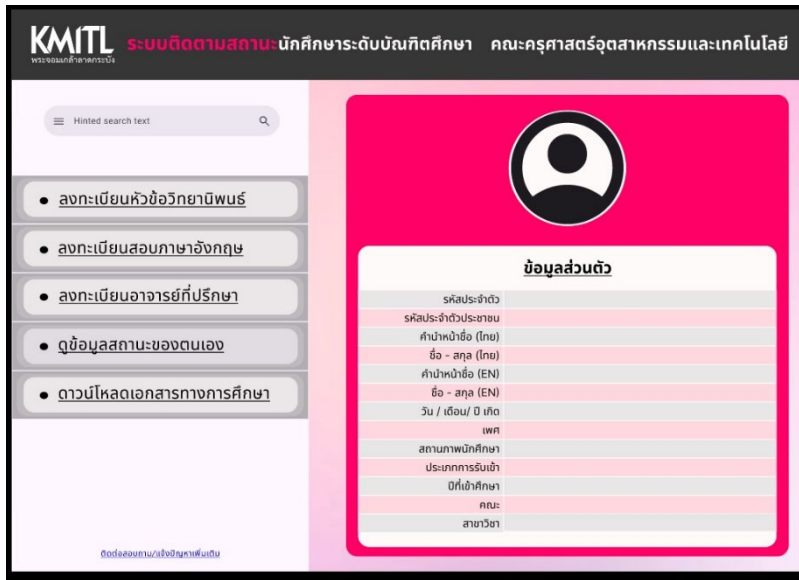


รูปที่ 1

-หน้าผู้ใช้งานหลัก



รูปที่ 2



รูปที่ 3

3.5. การแก้ไขแบบ UX/UI ของหน้าเว็บไซต์ และหน้า Report ที่ให้ user ใช้งาน

- Figma ที่ใช้ในการออกแบบ UX/UI

<https://shorturl.asia/uejOm>

4. แผนการปฏิบัติงานในสัปดาห์ต่อไป (Plans)

5. บันทึกเพิ่มเติม (Open Issues)

ลงชื่อ อาจารย์ที่ปรึกษา

วันที่