

โครงการเลขที่ วศ.คพ. P009-1/2568

เรื่อง

ระบบติดตามการพิจารณาหลักสูตร วิทยาลัยการศึกษาลดชีวิต มหาวิทยาลัยเชียงใหม่

โดย

นายจิรเดช อรทัย	รหัส 640610287
นายพนนพพันธุ์ ชนะดี	รหัส 640610645
นายทัตธน ศรีเงิน	รหัส 640612088
นางสาวสุรภา หลวงผิวเดช	รหัส 640612196

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ปีการศึกษา 2568

PROJECT No. CPE P009-1/2568

CMU Lifelong Education Assistance Website

Jiradete Oratai	640610287
Nontapan Chanadee	640610645
Thatthana Sringoen	640612088
Surapa Luangpiwdet	640612196

**A Project Submitted in Partial Fulfillment of Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering
Faculty of Engineering
Chiang Mai University
2025**

หัวข้อโครงการ : ระบบติดตามการพิจารณาหลักสูตร วิทยาลัยการศึกษาดลัดชีวิต มหาวิทยาลัยเชียงใหม่
: CMU Lifelong Education Assistance Website
โดย : นายจิรเดช อรทัย รหัส 640610287
นายณนทพันธุ์ ชนะดี รหัส 640610645
นายทัตธน ศรีเงิน รหัส 640612088
นางสาวสุรภา หลวงผิวเดช รหัส 640612196
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : ผศ.ดร. กำพล วรรดิษฐ์
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2568

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(ผศ.ดร. นวदनย์ คุณเลิศกิจ)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ
(ผศ.ดร. กำพล วรรดิษฐ์)

..... กรรมการ
(รศ.ดร. นริศรา เอี่ยมคณิตชาติ)

..... กรรมการ
(ผศ.ดร. ลัชนา ระมิงค์วงศ์)

หัวข้อโครงการ : ระบบติดตามการพิจารณาหลักสูตร วิทยาลัยการศึกษาลอตชีวิต มหาวิทยาลัยเชียงใหม่
: CMU Lifelong Education Assistance Website
โดย : นายจิรเดช อรทัย รหัส 640610287
นายณนทพันธุ์ ชนะดี รหัส 640610645
นายทัตธน ศรีเงิน รหัส 640612088
นางสาวสุรภา หลวงผิวเดช รหัส 640612196
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : ผศ.ดร. กำพล วรรดิษฐ์
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2568

บทคัดย่อ

“ระบบติดตามการพิจารณาหลักสูตร วิทยาลัยการศึกษาลอตชีวิต มหาวิทยาลัยเชียงใหม่” เป็นเว็บแอปพลิเคชันที่ออกแบบมาเพื่อช่วยเจ้าหน้าที่วิทยาลัยการศึกษาลอตชีวิต สามารถจัดการหลักสูตรที่ต้องการเปิดได้อย่างเป็นระบบ โดยเน้นการแก้ไขปัญหาในการเก็บข้อมูลตารางหลักสูตร ที่ไม่สามารถตอบสนองความต้องการได้มากพอ ซึ่งมักทำให้เกิดความสับสนในการจัดการหลักสูตรที่ต้องการเปิด และความยากลำบากในการติดตามสถานะของหลักสูตรที่ส่งเข้ามา

ผู้ใช้ในอดีตต้องใช้ระบบพูดคุยทางออนไลน์ เช่น ไลน์ ในการติดต่อสอบถามถึงสถานะของหลักสูตร ซึ่งอาจทำให้เจ้าหน้าที่ไม่สามารถตอบคำถามได้ทันที และอาจทำให้เกิดความล่าช้าในการพิจารณาหลักสูตร รวมถึงการโทรศัพท์สอบถามความคืบหน้าของหลักสูตรที่ส่งเข้ามา ซึ่งอาจทำให้เจ้าหน้าที่เสียเวลาในการทำงานมากขึ้น รวมถึงการนำข้อมูลนำใส่ในเว็บไซต์ของ วิทยาลัยการศึกษาลอตชีวิต มหาวิทยาลัยเชียงใหม่ ด้วยการคัดลอกข้อความจากไฟล์เอกสาร ที่ละบรรทัด ซึ่งอาจทำให้เกิดข้อผิดพลาดในการคัดลอกข้อมูลและเกิดความล่าช้าในการดำเนินการ

โครงการนี้จึงได้จัดทำเว็บไซต์ขึ้นเพื่อช่วยให้เจ้าหน้าที่สามารถสร้างหลักสูตรใหม่ได้ เห็นภาพรวมของหลักสูตรและดำเนินการต่อได้อย่างเป็นขั้นตอน อีกทั้งผู้ใช้ยังสามารถติดตามสถานะของหลักสูตรที่ส่งเข้ามาได้ผ่านเว็บไซต์โดยตรง

ผู้จัดทำหวังว่า “ระบบติดตามการพิจารณาหลักสูตร วิทยาลัยการศึกษาลอตชีวิต มหาวิทยาลัยเชียงใหม่” จะสามารถนำไปใช้งานได้จริง และช่วยลดปัญหาที่เกิดขึ้นในปัจจุบันได้อย่างมีประสิทธิภาพ

Project Title : CMU Lifelong Education Assistance Website
Name : Jiradete Oratai 640610287
Nontapan Chanadee 640610645
Thatthana Sringoen 640612088
Surapa Luangpiwdet 640612196
Department : Computer Engineering
Project Advisor : Asst.Prof. Kampol Woradit, Ph.D.
Degree : Bachelor of Engineering
Program : Computer Engineering
Academic Year : 2025

ABSTRACT

The "CMU Lifelong Education Assistance Website" is a web application designed to assist the staff of the College of Lifelong Education in systematically managing proposed courses. The system aims to address the existing issues in maintaining course-related data tables, which have proven insufficient in meeting operational needs—often resulting in confusion when handling course proposals and difficulties in tracking their review status.

In the past, users relied on online communication platforms such as Line to inquire about the status of their course proposals. This method often caused delays, as staff members were unable to provide immediate responses, consequently slowing down the course evaluation process. Additionally, inquiries made through phone calls increased the workload and consumed valuable working time. Another challenge was the manual process of transferring course information to the College's official website by copying text from document files line by line, which frequently led to data entry errors and further delays in operations.

To solve these problems, this project developed a website that enables staff members to create new course entries, view a comprehensive overview of all courses, and manage each stage of the process more systematically. Meanwhile, users can directly track the status of their submitted courses through the website.

The developers hope that the "CMU Lifelong Education Assistance Website" will be successfully implemented and effectively reduce the existing problems faced by both the College and its users.

กิตติกรรมประกาศ

โครงการนี้จะไม่สามารถเกิดขึ้นได้เลยหากขาดความร่วมมือจากทุกภาคส่วน โดยเฉพาะคณาจารย์ และอาจารย์ที่ปรึกษา ผศ.ดร. กำพล วรรดิษฐ์ ที่คอยให้คำปรึกษา เทคนิค และแนะแนวทางในการพัฒนาโครงการนี้จนสามารถบรรลุวัตถุประสงค์ได้

ขอขอบคุณสถาบัน Lifelong Education ที่ให้โอกาสในการพัฒนาโครงการนี้ รวมถึงการประชุมปรึกษาและให้คำแนะนำอย่างต่อเนื่อง ทำให้โครงการนี้สำเร็จได้สมบูรณ์ และขอขอบคุณ รศ.ดร. นริศรา เอี่ยมคณิตชาติ และ ผศ.ดร. ถังนา ระมิงค์วงศ์ ผู้เป็นกรรมการสอบโครงการที่ได้ให้คำแนะนำอันมีคุณค่า

นอกจากนี้ ขอขอบคุณ Entaneer Academy ที่ช่วยให้พวกเราเข้าใจระบบ Lifelong Education ได้ลึกซึ้งมากยิ่งขึ้น จนสามารถพัฒนาโครงการนี้ได้จนถึงเวอร์ชันปัจจุบัน

สุดท้ายนี้ ขอขอบคุณสมาชิกโครงการทุกคนที่ได้ทุ่มเทแรงกาย แรงใจ และเวลา เพื่อให้โครงการนี้สำเร็จ ลุล่วง ขอขอบคุณทุกท่านจากใจจริง

นายจิรเดช อรทัย

นายณนทพันธุ์ ชนะดี

นายทัตธน ศรีเงิน

นางสาวสุรภา หลวงผิวเดช

4 ตุลาคม 2568

สารบัญ

บทคัดย่อ	ข
Abstract	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญรูป	ช
สารบัญตาราง	ณ
1 บทนำ	1
1.1 ที่มาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.3.1 ขอบเขตด้านข้อมูล	2
1.3.2 ขอบเขตด้านฮาร์ดแวร์	2
1.3.3 ขอบเขตด้านซอฟต์แวร์	2
1.4 ประโยชน์ที่ได้รับ	2
1.5 เทคโนโลยีและเครื่องมือที่ใช้	2
1.5.1 เทคโนโลยีด้านฮาร์ดแวร์	2
1.5.2 เทคโนโลยีด้านซอฟต์แวร์	3
1.6 แผนการดำเนินงาน	3
1.7 บทบาทและความรับผิดชอบ	4
1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม	4
2 ทฤษฎีที่เกี่ยวข้อง	6
2.1 Design	6
2.1.1 Database Design	6
2.1.2 Hexagonal Architecture	7
2.2 UX/UI Design	9
2.2.1 UX Design (User Experience Design)	9
2.2.2 UI Design (User Interface Design)	10
2.3 Next.js with TypeScript	10
2.3.1 Next.js	10
2.3.2 TypeScript	11
2.3.3 Next.js + TypeScript	11
2.4 Go programming language	12
2.4.1 หลักการทางทฤษฎีที่สำคัญในการออกแบบภาษา Go	12
2.5 JSON	13
2.6 Pandoc	14
2.7 Regular expression	14
3 โครงสร้างและขั้นตอนการทำงาน	16
3.1 การสำรวจและทำความเข้าใจปัญหา	16
3.1.1 สัมภาษณ์และวิเคราะห์ผู้ใช้งาน	16
3.1.2 สรุปปัญหาและตั้งวัตถุประสงค์ของโครงการ	16
3.1.3 แผนภาพการดำเนินการส่งหลักสูตร	16
3.2 User interface (UI) and Design	16

3.2.1	User flow	16
3.2.2	Role	17
3.2.3	Design System	17
3.2.4	User Interface Design	17
3.3	หลักการทำงานของเว็บแอปพลิเคชัน	18
3.3.1	หลักการทำงานของเว็บไซต์	18
3.3.2	แดชบอร์ดหลัก	18
3.4	การมองเห็นและการใช้งานแต่ละบทบาท	22
3.4.1	บทบาท Admin	22
3.4.2	บทบาท Staff	22
3.4.3	บทบาท LE	22
3.4.4	บทบาท Officer	22
3.4.5	บทบาท User	22
3.4.6	กรอกข้อมูลอัตโนมัติจากไฟล์เอกสาร	22
3.5	การออกแบบฐานข้อมูล	23
3.5.1	โครงสร้างฐานข้อมูล	23
3.6	API endpoint dashboard	26
3.6.1	การตั้งค่า Middleware JWT	26
3.7	รายละเอียดการกำหนดเส้นทางตามบทบาท (Role-Based Routing)	26
3.7.1	Shared Routes (บทบาท: admin, staff, LE, officer, user)	26
3.7.2	Officer Routes (บทบาท: admin, staff, LE, officer)	26
3.7.3	LE Routes (บทบาท: admin, staff, LE)	26
3.7.4	Staff Routes (บทบาท: admin, staff)	27
3.7.5	Admin Routes (บทบาท: admin)	27
3.8	Course Document Converter	28
3.9	API endpoint document converter	30
4	การทดลองและผลลัพธ์	31
4.1	การทดสอบการทำงานของระบบ	31
4.1.1	วัตถุประสงค์ในการทดสอบ	31
4.1.2	สภาพแวดล้อมของการทดสอบ (System Environment)	31
4.1.3	ขั้นตอนและกระบวนการทดสอบ	32
4.2	การทดสอบการใช้งานจริงของผู้ใช้	32
4.2.1	การออกแบบการทดสอบ	32
4.2.2	ผลการประเมินความพึงพอใจของผู้ใช้งาน	32
4.2.3	ผลการประเมิน Net Promoter Score (NPS)	33
4.2.4	การรวบรวมความคิดเห็นจากผู้ใช้งาน (Feedback)	33
4.2.5	สรุปผลการทดลอง	34
5	บทสรุปและข้อเสนอแนะ	35
5.1	สรุปผล	35
5.2	ปัญหาที่พบและแนวทางการแก้ไข	35
5.2.1	การแปลงไฟล์เอกสารบางรูปแบบไม่สมบูรณ์	35
5.2.2	การขาดระบบแจ้งเตือนอัตโนมัติ (Notification System)	35
5.2.3	ข้อจำกัดด้านเวลาการเก็บข้อมูลทดสอบจริง	35
5.3	ข้อเสนอแนะและแนวทางการพัฒนาต่อ	35

บรรณานุกรม	37
ก การเก็บข้อมูลเพื่ออ้างอิงกับระบบ	39
ก.1 Flow การทำงานของ Entaneer Academy	39
ก.2 Flow การทำงานของ วิทยาลัยการศึกษาดลิตชีวิต	39
ก.3 การได้รับ feedback หลักจากให้ผู้ใช้งานทดลองใช้ระบบ	39
ข คู่มือการใช้งานระบบ	41
ข.1 Login	41
ข.2 สร้างรายการหลักสูตรใหม่	41
ข.3 การแก้ไขหลักสูตร	41
ข.4 เพิ่ม อัปเดต ลบ ผู้ใช้ในระบบ	41
ข.5 สร้าง ลบ เปลี่ยนชื่อ สถานะเจ้าหน้าที่	41
ข.6 สร้าง ลบ เปลี่ยนชื่อ รายการที่ต้องทำแต่ละรายการหลักสูตร	41
ข.7 กรองสถานะ และการค้นหารายการหลักสูตร	41
ประวัติผู้เขียน	42

สารบัญรูป

2.1	ขั้นตอนการออกแบบฐานข้อมูล	6
2.2	โครงสร้าง Hexagonal Architecture เบื้องต้น	7
2.3	กระบวนการทำงานของ Hexagonal Architecture	9
2.4	กระบวนการทำงานของ Go ที่แปลงเป็น Executable	13
2.5	ตัวอย่างสัญลักษณ์ Metacharacter ที่ใช้ใน Regular Expression	15
3.1	User Flow ของระบบ	17
3.2	figma ในการออกแบบหน้าตาระบบ	18
3.3	หน้า Login	18
3.4	หน้า Dashboard	19
3.5	หน้าสร้างหลักสูตรใหม่	19
3.6	หน้าแก้ไขหลักสูตรที่สร้าง	20
3.7	หน้าฟังก์ชันในการกรอกรายการหลักสูตรด้วย course status	20
3.8	หน้าฟังก์ชันค้นหารายการหลักสูตรด้วยชื่อหลักสูตร	20
3.9	หน้าสำหรับจัดการผู้ใช้ในระบบ	21
3.10	หน้าสำหรับจัดการสถานะของเจ้าหน้าที่	21
3.11	หน้าสำหรับจัดการรายการที่ต้องทำแต่ละหลักสูตร	22
3.12	ERdiagram สำหรับออกแบบระบบ	23
ก.1	กระบวนการทำงานของ Entaneer Academy ในการส่งหลักสูตร	39

สารบัญตาราง

4.1	ผลการประเมินความพึงพอใจของผู้ใช้งาน	33
4.2	ผลการประเมิน Net Promoter Score (NPS)	33

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

ในปัจจุบัน วิทยาลัยการศึกษาลอตซีวิต มหาวิทยาลัยเชียงใหม่ มีการเปิดรับหลักสูตรทั้งใหม่และเก่า ที่คณาจารย์ต้องการที่จะเปิดสอน เพื่อให้ผู้เรียนทุกช่วงวัยได้ศึกษา ซึ่งจะมีปัญหาในกระบวนการทำงานที่ทำให้ล่าช้า เช่น ตารางข้อมูลหลักสูตรที่จัดการได้ยากในการจัดการ และการติดตามสถานะการพิจารณาหลักสูตรที่ไม่สามารถทำได้อย่างมีประสิทธิภาพ และ การคัดลอกข้อมูลหลักสูตรจากไฟล์เอกสารเพื่อไปลงในเว็บไซต์ของวิทยาลัยการศึกษาลอตซีวิตที่ต้องทำที่ละบรรทัด ซึ่งอาจทำให้เกิดข้อผิดพลาดในการกรอกข้อมูลและล่าช้าในการดำเนินงานได้ พวกเราได้เล็งเห็นปัญหาในส่วนนี้ จึงต้องการสร้างเว็บไซต์ที่สามารถช่วยแก้ไขปัญหาดังกล่าวได้โดยสามารถจัดการข้อมูลหลักสูตรได้ตามที่ต้องการ และสามารถมองเห็นภาพรวมได้มากขึ้น สามารถให้ผู้ใช้งานสามารถติดตามสถานะของหลักสูตรที่ตนเองได้ส่งไปได้โดยตรงผ่านทางเว็บไซต์ และให้เจ้าหน้าที่ของวิทยาลัยสามารถเพิ่มข้อมูลหลักสูตร จากไฟล์เอกสารเข้าสู่ระบบได้ทันที

1.2 วัตถุประสงค์ของโครงการ

1. การจัดสรรข้อมูลหลักสูตรภายในระบบ

- สามารถรวบรวมและจัดเก็บข้อมูลหลักสูตรให้เป็นระเบียบ ง่ายต่อการเข้าถึงและค้นหา
- สามารถสร้างคิวในระบบ ทำให้สามารถมองเห็นภาพรวมของคิวได้ และตัดสินใจได้ว่าจะเลือกทำอะไรก่อน
- รองรับการปรับแต่งและอัปเดตข้อมูล เพิ่ม/ลด รายละเอียดที่เกี่ยวข้องกับหลักสูตรได้อย่างยืดหยุ่น

2. การติดตามความคืบหน้าของหลักสูตร

- ผู้ใช้งานสามารถตรวจสอบสถานะการเรียนรู้ ความก้าวหน้า และต้องการที่จะแก้ไขหลักสูตร

3. การจัดการสิทธิ์และบทบาทผู้ใช้งาน

- กำหนดสิทธิ์การเข้าถึงข้อมูลและฟังก์ชันการใช้งานของระบบอย่างเหมาะสม เช่น Admin, Staff, LE, Officer, User
- จำกัดสิทธิ์การมองเห็นหรือการแก้ไขข้อมูลเพื่อความปลอดภัยและความถูกต้องของข้อมูล

4. การเพิ่มข้อมูลหลักสูตรจากไฟล์เอกสาร

- สามารถเพิ่มข้อมูลหลักสูตรจากไฟล์เอกสารเข้าสู่ระบบได้ทันที
- ลดข้อผิดพลาดในการกรอกข้อมูลและล่าช้าในการดำเนินงาน

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตด้านข้อมูล

โครงการนี้ใช้ข้อมูลจาก

1. วิทยาลัยการศึกษาลอตซีวิต มหาวิทยาลัยเชียงใหม่

- ข้อมูลกระบวนการทำงานของวิทยาลัยการศึกษาลอตซีวิต ในการเปิดหลักสูตร และหน้าตาพื้นที่การเก็บข้อมูลหลักสูตร
- ข้อมูลระบบเว็บไซต์ของวิทยาลัยการศึกษาลอตซีวิต ในการเห็นกระบวนการเปิดหลักสูตรผ่านเว็บไซต์

2. Entaneer Academy

- ข้อมูลกระบวนการทำงานของเจ้าหน้าที่ของคณะ ในการส่งข้อมูลหลักสูตร และกระบวนการภายในคณะ

1.3.2 ขอบเขตด้านฮาร์ดแวร์

ผู้ใช้จะใช้แล็ปท็อประบบปฏิบัติการวินโดวส์ และ แล็ปท็อประบบปฏิบัติการไอโอเอสได้

1.3.3 ขอบเขตด้านซอฟต์แวร์

จำกัดเฉพาะผู้ที่มีบัญชี CMU Account ในสถานะ อาจารย์ เจ้าหน้าที่ภาควิชาของแต่ละคณะ และ เจ้าหน้าที่ของวิทยาลัยการศึกษาลอตซีวิต เท่านั้นจึงจะสามารถเข้าใช้งานได้ และสามารถใช้งานผ่านเว็บเบราว์เซอร์ได้เท่านั้น

1.4 ประโยชน์ที่ได้รับ

- สามารถจัดสรรและจัดการคำร้องในการเปิดหลักสูตรได้อย่างมีประสิทธิภาพมากขึ้น
- สามารถติดตามสถานะของหลักสูตรที่ตนเองได้ส่งไปได้โดยตรงผ่านทางเว็บไซต์
- สามารถลดเวลาในการกรอกข้อมูลจากไฟล์เอกสารเข้าสู่ระบบทางวิทยาลัยการศึกษาลอตซีวิตได้ทันที

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านฮาร์ดแวร์

เครื่องคอมพิวเตอร์ตั้งโต๊ะ แล็ปท็อป บนหน้าจอที่มีความละเอียด 1152 x 854 พิกเซล ขึ้นไป

1.5.2 เทคโนโลยีด้านซอฟต์แวร์

Frontend

- React : JavaScript library สำหรับสร้าง User Interface
- Tailwind CSS : Utility CSS framework ที่ช่วยพัฒนาสร้าง User Interface
- Figma : Web Application สำหรับ ออกแบบ User Interface และ User Experience

Backend

- GoFiber : Go Web Framework ที่สร้างอยู่บน Fasthttp ซึ่งเป็น library จัดการ HTTP server ซึ่งเป็น HTTP engine
- Postman : Application สำหรับการพัฒนาและทดสอบ API services
- Pandoc : เครื่องมือสำหรับแปลงเอกสาร (Document Converter) แบบ open-source ที่สามารถแปลงไฟล์ได้หลากหลายรูปแบบ

Database

- PostgreSQL : เป็นระบบจัดการฐานข้อมูลแบบ object-relational database management system แบบ OpenSource ที่ใช้ในการเก็บข้อมูลเกี่ยวกับหลักสูตรและโครงสร้างของระบบ

Infrastructure

- Docker : Infrastructure manager ใช้สำหรับการสร้างหรือจัดการ service ต่างๆ
- Nginx : HTTP server แบบ OpenSource
- GithubAction : CI/CD platform ที่ใช้สำหรับการอัตโนมัติในการ build และ deploy code

Third party

- OAuth with EntraID : เป็น Industry-standard protocol ใช้สำหรับการทำ Authorization framework เปิดให้third-party application ได้รับสิทธิการเข้าถึง

1.6 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	มี.ย. 2568	ก.ค. 2568	ส.ค. 2568	ก.ย. 2568	ต.ค. 2568
สำรวจปัญหาเบื้องต้น					
กำหนดปัญหา					
กำหนดแนวทางแก้ปัญหา และวัตถุประสงค์					

ขั้นตอนการดำเนินงาน	มิ.ย. 2568	ก.ค. 2568	ส.ค. 2568	ก.ย. 2568	ต.ค. 2568
UX/UI ,Bussiness logic, Database design					
สำรวจปัญหาเพิ่มเติมจากวิทยาลัยการศึกษาดลอดชีวิต					
กำหนดขอบเขตของ Users					
UX/UI, Frontend, Bussiness logic, Database Implement V.1					
Testing Bussiness logic and Database V.1					
UI and Bussiness logic Feedback Update 1					
UX/UI, Bussiness logic, Database Implement V.2					
Testing Bussiness logic and Database V.2					
UI and Bussiness logic Feedback Update 2					
UX/UI, Bussiness logic, Database Implement V.3					
Testing Bussiness logic and Database V.3					
สำรวจปัญหาเพิ่มเติมจาก Entaneer Academy					
Finalized Frontend and Backend					
Deploy Final Web Application					
นำไปให้ User ทดลองใช้เพื่อรับ Feedback และการประเมินผล					
เขียนรายงาน Final Project และเตรียม Slide ในการนำเสนอออนไลน์					

1.7 บทบาทและความรับผิดชอบ

1. นายจิรเดช อรทัย มีบทบาทเป็น Backend Developer (Feature 1 : แดชบอร์ดสำหรับแอดมิน และผู้ใช้) โดยรับผิดชอบในส่วน Backend Logic , API รวมถึงการออกแบบโครงสร้างข้อมูล (Data Structure Design) และการทดสอบการใช้งาน
2. นายนันทพันธ์ุ ชนะดี มีบทบาทเป็น Backend Developer (Feature 2 : เติมข้อมูลอัตโนมัติ) และ Co-Frontend Developer โดยรับผิดชอบในส่วน Backend Logic และ API รวมถึง Co-Frontend Developer
3. นายทัตชน ศรีเงิน มีบทบาทเป็น Frontend Developer โดยรับผิดชอบในส่วน Frontend
4. นางสาวสุรภา หลวงผิวดเดช มีบทบาทเป็น UX/UI Designer โดยรับผิดชอบในส่วนการออกแบบ UX/UI ที่เหมาะสมกับผู้ใช้ และ ทดสอบการใช้งาน

1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

การที่เจ้าหน้าที่ของวิทยาลัยการศึกษาดลอดชีวิตสามารถทำงานในสภาพแวดล้อมที่เอื้ออำนวยต่อการทำงาน พร้อมกับลดปัญหาการสอบถามสถานะที่ทำให้สูญเสียเวลาในการดำเนินงาน รวมถึง สามารถจัดการข้อมูลจากเอกสารเข้าสู่ระบบได้ทันที และในส่วนของผู้ใช้สามารถวางใจกับการดำเนินการของหลักสูตรโดยสามารถ

ดูจากสถานะของหลักสูตรได้ ซึ่งช่วยให้ผู้ใช้ ไม่ว่าจะเป็น เจ้าหน้าที่ของวิทยาลัยการศึกษาดลัดชีวิต หรือผู้ใช้
เกิดความมั่นใจ และลดความกังวลในการติดตามสถานะของหลักสูตรและการกรอกข้อมูลหลักสูตรสู่ระบบ

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

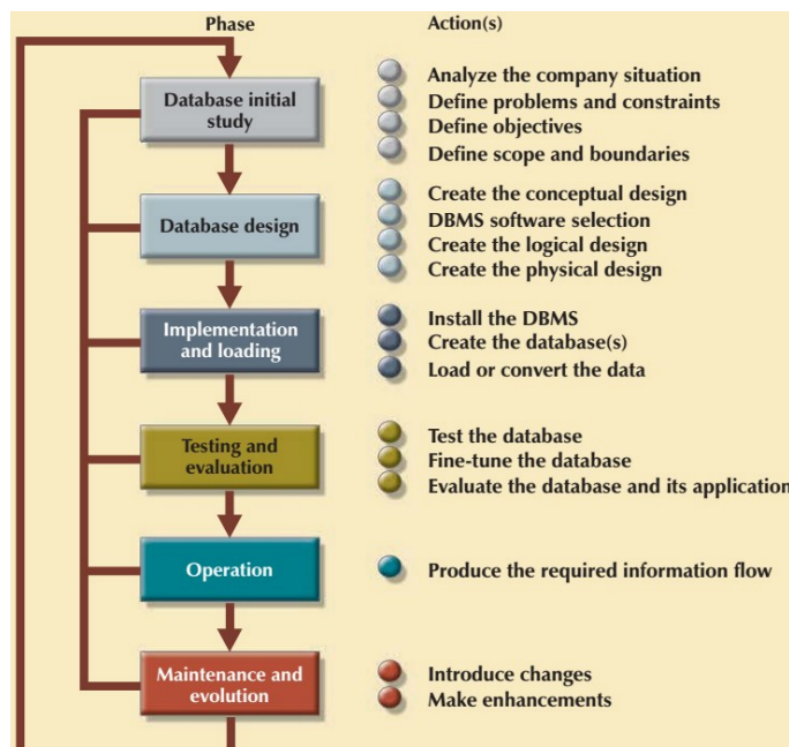
ในบทความนี้จะอธิบายเกี่ยวกับทฤษฎีและเครื่องมือที่เกี่ยวข้องในการทำงานให้ประสบความสำเร็จและสมบูรณ์ตามวัตถุประสงค์

2.1 Design

2.1.1 Database Design

[9]ภายใต้การออกแบบระบบเทคโนโลยีสารสนเทศ การออกแบบฐานข้อมูลจะเป็นขั้นตอนหนึ่งที่เรา จะต้องให้ความสำคัญ ขั้นตอนการออกแบบฐานข้อมูลจะประกอบไปด้วย 6 ขั้นตอนหลัก ที่จะประกอบด้วย

1. การศึกษาฐานข้อมูลขององค์กรเบื้องต้น (database initial study)
2. การออกแบบฐานข้อมูล (database design)
3. การสร้างฐานข้อมูลและการโหลดข้อมูลเข้าสู่ฐานข้อมูล (implementation and loading)
4. การทดสอบและประเมินการทำงานของฐานข้อมูล (testing and evaluation)
5. การประยุกต์ใช้ฐานข้อมูล (operation)
6. การดูแลรักษาและพัฒนาให้ดียิ่งขึ้น (maintenance and evolution)

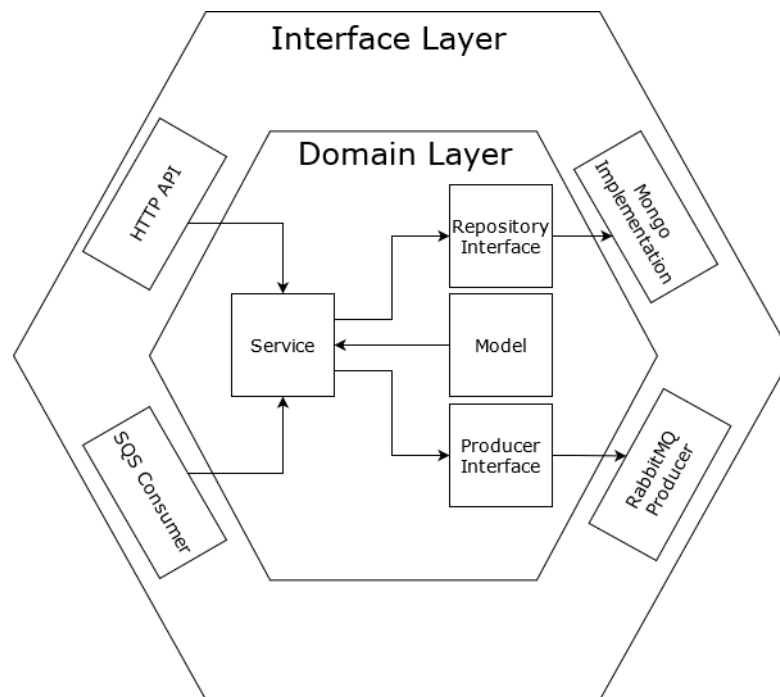


รูปที่ 2.1: ขั้นตอนการออกแบบข้อมูล

2.1.2 Hexagonal Architecture

Hexagonal architecture[6] ถูกคิดค้นโดย Alistair Cockburn เมื่อปี 2005 โดยเป็น 1 ใน Code architecture pattern ที่เป็นที่นิยมของ Go ที่สร้างอยู่บน 2 pattern ใหญ่ๆคือ Adaptor Pattern และ Dependency Injection โดย

1. Hexagonal จะนำเสนอส่วนที่เป็น "business logic" ขึ้นมาโดยไม่สนใจว่า ส่วนที่มาเชื่อมต่อจะเป็นส่วนไหน ขอแค่ทำตาม spec ของ business logic = สามารถใช้งานได้แล้ว
2. เป็นการแยกส่วนกันระหว่าง inside part (business logic, domain layer) และ outside part (ส่วนเชื่อมต่อกับภายนอกเช่น UI, database, external service - interface layer)



รูปที่ 2.2: โครงสร้าง Hexagonal Architecture เบื้องต้น

โดยองค์ประกอบใหญ่ๆของ Hexagonal Architecture จะประกอบด้วยของ 3 อย่างใหญ่ๆคือ

1. Ports ส่วน interfaces ที่บอกว่า application สามารถเชื่อมต่อมายัง business logic ได้ยังไง หรือสามารถเข้าถึง resource ภายนอกได้ยังไง
2. Adapters ส่วนที่จะมาเชื่อมต่อกับ Port เป็นเหมือน "สะพาน" ระหว่าง Resource จริง กับ business logic (เช่น เป็นคนคุยกับ database, web service เป็นต้น)
3. Domain-centric ส่วนตรงกลางของ Business logic เป็นเหมือนศูนย์กลางของการคำนวณ

โดยแต่ละส่วนมีรายละเอียดดังนี้

1. Port คือ concept ของการแยกส่วนระหว่าง application's core logic (the domain) ออกจากส่วนที่ "สามารถกระทำได้จากภายนอก (external)" เช่น user interface, database

- Port ทำหน้าที่เป็นตัวแทนของการสื่อสารแกนกลาง (Business logic) และจะทำหน้าที่ส่งต่อไปยัง business logic
- ในทาง programming Port จะถูกเรียกว่า "interface" หมายถึงส่วนที่ใช้สำหรับแลกเปลี่ยนข้อมูล

โดยปกติ Port จะมี 2 ประเภท

- (a) Primary or Driving Ports เป็น interfaces ที่ถูกใช้งานจากภายนอก เช่น Web API (ที่ client application เรียกใช้)
- (b) Secondary or Driven Ports เป็น interface ที่ใช้สำหรับจัดการกับ Resource ภายนอก ซึ่งปกติจะเกี่ยวกับการจัดการ external service หรือ database

ยกตัวอย่างสำหรับ 2 เคสนี้ สมมติว่าเราต้องทำ API สำหรับสร้าง Order ขึ้นมา โดยเราจะแยกออกเป็น 2 ส่วนคือ

- (a) Primary Ports = ส่วน Web API ที่ใช้สำหรับรับข้อมูล order จาก user (จาก class diagram ด้านล่างคือ OrderServiceInterface)
- (b) Secondary Ports = ส่วน interface ที่เชื่อมต่อไปยัง database เพื่อบันทึกข้อมูล order (จาก class diagram ด้านล่างคือ OrderRepositoryInterface)

2. Adapter คือ component ที่เชื่อมต่อระหว่าง application core logic (ผ่าน Port) กับส่วนของ External Resource

- adapter เป็นส่วนสำคัญที่ทำหน้าที่เป็นเหมือน "สะพาน" เชื่อมต่อระหว่าง application และ external component
- เป็นคนคอยแปลง request และส่ง data ให้

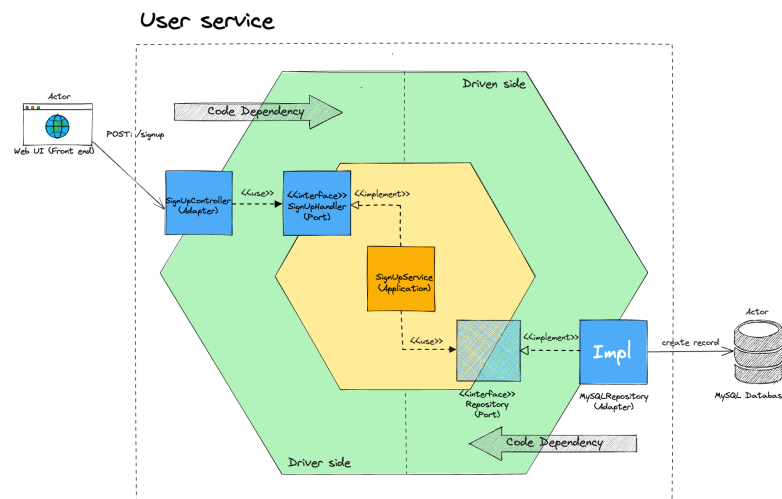
โดยปกติ Adapter มี 2 ประเภทเช่นกัน

- (a) Primary or Driving Adapters adapter สำหรับการ handle (และทำการแปลง) input ที่เข้ามาใน application เพื่อเตรียมสำหรับใช้งานใน application core logic เช่น Web Controller, CLI (Command Line Interface) มี OrderHandler ทำหน้าที่เป็น HTTP handler ที่รับข้อมูลจาก web server โดยโอเดียคือ OrderHandler จะรับข้อมูลจาก HTTP request มาเพื่อใช้สำหรับการสร้าง order โดยทำการดึง data ออกจาก request และทำการแปลงข้อมูลไปใน format ที่ core application สามารถใช้งานได้
- (b) Secondary or Driven Adapters adapter สำหรับการพูดคุยกับภายนอก เช่น database adapter สำหรับการแปลงข้อมูลเป็น SQL query เพื่อใช้สำหรับเรียกไปยัง database ปลายทาง มี GormOrderRepository ทำหน้าที่เป็น database adapter โดยมีหน้าที่นำข้อมูล order จาก application's core logic และทำการแปลงเป็นคำสั่งสำหรับการส่งเข้า Database (เช่น แปลงเป็นคำสั่ง SQL)

โดยทั่วไปแล้ว การ implement adapter จะ implement ตาม port interface ที่ถูกกำหนดจาก application's core เพื่อเป็นการแยกส่วนกันระหว่าง application internal logic และ การพูดคุยกับภายนอกออกมา

3. Application core (หรือหลายๆที่จะเรียกว่า Domain Logic หรือ Business Logic) คือส่วนตรงกลางของ architecture ซึ่งเป็น logic หลักของระบบที่เหมือนเป็น "กฎ" การทำงานของระบบนี้

- เป็นส่วนที่กำหนดรูปแบบของ data structure ที่ใช้งานภายใน application นี้
- เป็นส่วนที่เก็บ function และ operation หลักของ application
- มีส่วน orderServiceImpl สำหรับการ implement OrderService โดยเป็นการ implement ส่วนของ logic CreateOrder ว่าตอนสร้าง order ต้องจัดการ data ยังไงบ้าง
- ช่น อาจจะเพิ่มการ validate data มาว่ารับข้อมูลชุดนี้หรือไม่, ถ้ามีสินค้าไม่ถึงจะทำการแสดง Error ออกไป เป็นต้น



รูปที่ 2.3: กระบวนการทำงานของ Hexagonal Architecture

2.2 UX/UI Design

2.2.1 UX Design (User Experience Design)

มุ่งเน้นไปที่ประสบการณ์โดยรวมของผู้ใช้ที่มีต่อ software องค์ประกอบหลักของ UX Design

- การใช้งาน: ผู้ใช้สามารถใช้งานผลิตภัณฑ์ได้ง่ายและสะดวกหรือไม่
- ความพึงพอใจ: ผู้ใช้รู้สึกอย่างไรกับการใช้งานผลิตภัณฑ์
- ประโยชน์: ผลิตภัณฑ์ตอบสนองความต้องการของผู้ใช้หรือไม่

หลักการออกแบบ UX

- การวิจัยผู้ใช้: เข้าใจความต้องการ พฤติกรรม และเป้าหมายของผู้ใช้
- การสร้าง prototype: ทดสอบและปรับแต่งการออกแบบ UX
- การวัดผล: วิเคราะห์ประสิทธิภาพของ UX Design

2.2.2 UI Design (User Interface Design)

UI Design มุ่งเน้นไปที่การออกแบบส่วนติดต่อผู้ใช้(UI) ของ software องค์ประกอบหลักของ UI Design

- Layout: การจัดวางองค์ประกอบต่างๆ บนหน้าจอ
- Design: การใช้สี รูปภาพ และ Graphic
- Interactive: การออกแบบวิธีการที่ผู้ใช้โต้ตอบกับ Web-app

หลักการออกแบบ UI

- Wireframe Design: ออกแบบโครงสร้างพื้นฐานของ UI
- Visual Design: ออกแบบรูปลักษณ์ของ UI
- User Test: ทดสอบ UI กับผู้ใช้

2.3 Next.js with TypeScript

2.3.1 Next.js

Next.js[8] เป็น Framework ของ React ที่ถูกพัฒนาโดยบริษัท Vercel เพื่อช่วยให้การพัฒนาเว็บแอปพลิเคชันด้วย React มีประสิทธิภาพมากขึ้น และสามารถทำงานได้ทั้ง ฝั่ง Frontend และ Backend ในโปรเจกต์เดียวกัน โดย Next.js ถูกออกแบบมาเพื่อแก้ปัญหาการเรนเดอร์หน้าเว็บ การจัดการเส้นทาง (Routing) และการเพิ่มประสิทธิภาพของเว็บไซต์ เช่น SEO และ Performance ที่ React เพียงอย่างเดียวไม่รองรับโดยตรง

Next.js ใช้ภาษาหลักคือ JavaScript หรือ TypeScript เหมือนกับ React และยังรองรับการเขียนด้วย JSX / TSX เช่นเดียวกัน แต่จะเพิ่มเติมโครงสร้างและฟีเจอร์ระดับ Framework เข้ามา เพื่อให้สามารถจัดการโครงสร้างหน้าเว็บ การโหลดข้อมูล และการเรนเดอร์ได้ง่ายขึ้น

แนวคิดหลักของ Next.js

1. File-based Routing: Next.js มีระบบ Routing อัตโนมัติ โดยจะสร้างเส้นทางของหน้าเว็บจากโครงสร้างของโฟลเดอร์และไฟล์ในโปรเจกต์ ทำให้ไม่จำเป็นต้องใช้ไลบรารี react-router เหมือน React ทั่วไป
2. Rendering Modes: Next.js รองรับการเรนเดอร์หลายรูปแบบ ได้แก่
 - SSR (Server-Side Rendering): เรนเดอร์หน้าเว็บจากฝั่งเซิร์ฟเวอร์ก่อนส่งให้ผู้ใช้ เหมาะกับเว็บไซต์ที่ต้องการ SEO สูง
 - SSG (Static Site Generation): สร้างหน้าเว็บแบบคงที่ตั้งแต่ตอน build เหมาะกับเว็บที่ข้อมูลไม่เปลี่ยนแปลงบ่อย เช่น Blog

- CSR (Client-Side Rendering): เรนเดอร์ผ่านฝั่ง Client แบบ React ปกติ
 - ISR (Incremental Static Regeneration): อัปเดตหน้าเว็บแบบอัตโนมัติหลังจากสร้างไฟล์ static ไว้แล้ว
3. API Routes Next.js สามารถสร้าง API ได้ในตัวเอง โดยเขียนในไฟล์เตอร์ `app/api` หรือ `pages/api` ทำให้สามารถสร้าง Backend เล็กๆภายในโปรเจกต์เดียวกับ Frontend
 4. Optimizations & SEO Friendly Next.js มีระบบเพิ่มประสิทธิภาพในตัว เช่น
 - Image Optimization ผ่าน `<Image />` component
 - การจัดการ Metadata และ Head อัตโนมัติ
 - การแยกโค้ด (Code Splitting) อัตโนมัติ
 - รองรับ TypeScript โดยสมบูรณ์
 5. Component-based Development เช่นเดียวกับ React, Next.js ยังคงใช้แนวคิด Component เป็นหลัก โดยแต่ละส่วนของหน้าเว็บ เช่น Navbar, Footer, หรือ Content ถูกสร้างแยกออกจากกัน เพื่อให้สามารถนำกลับมาใช้ซ้ำได้

2.3.2 TypeScript

TypeScript[3] คือ Super Set ของ JavaScript ที่จุดเด่นคือถูกพัฒนาให้มี Strongly typed ทำให้ต้องระบุ Type ของตัวแปรให้ชัดเจนก่อนจึงจะสามารถเรียกใช้งานและรันโค้ดต่อไปได้ ซึ่งจะช่วยให้โปรแกรมมีความแน่นอนมากขึ้น ผู้พัฒนาสามารถพัฒนาโปรเจกต์ได้อย่างมีประสิทธิภาพ ทำความเข้าใจได้ง่าย และสามารถป้องกันความผิดพลาดหรือช่องโหว่ของ JavaScript's error เมื่อทำการ compile

2.3.3 Next.js + TypeScript

การใช้ Next.js ร่วมกับ TypeScript ช่วยเพิ่มประสิทธิภาพ ความปลอดภัย และความสามารถในการขยาย (scalability) ของซอฟต์แวร์ได้อย่างมาก เนื่องจาก Next.js เป็น Framework ที่ต่อยอดมาจาก React ซึ่งรองรับ TypeScript โดยสมบูรณ์ในตัว ทำให้ผู้พัฒนาสามารถสร้างเว็บแอปพลิเคชันที่มีโครงสร้างชัดเจน มีการตรวจสอบชนิดข้อมูล (type checking) อย่างเข้มงวด และลดความผิดพลาดที่อาจเกิดขึ้นในระหว่างการพัฒนาได้

TypeScript ช่วยให้การเขียนโค้ดใน Next.js มีความ อ่านง่าย (readable), เข้าใจง่าย (maintainable) และสามารถ ตรวจสอบข้อผิดพลาดในขณะคอมไพล์ (compile-time errors) ได้ก่อนที่จะรันจริง ซึ่งช่วยให้โค้ดมีความเสถียรและปลอดภัยมากขึ้น นอกจากนี้ การใช้ TypeScript ยังทำให้ทีมพัฒนาสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ เพราะตัวระบบสามารถบอกชนิดของข้อมูล ฟังก์ชัน และ props ของ component ได้อย่างชัดเจน

ด้วยเหตุนี้ การใช้ Next.js + TypeScript จึงเหมาะสำหรับการพัฒนาเว็บแอปพลิเคชันขนาดกลางถึงขนาดใหญ่ที่มีความซับซ้อน หรือระบบที่ต้องการความแม่นยำในการจัดการข้อมูล เช่น Dashboard, ระบบจัดการข้อมูล (CMS) หรือเว็บองค์กรที่ต้องการความปลอดภัยสูง

2.4 Go programming language

ภาษาโปรแกรม Go[7] คือภาษาโปรแกรมแบบโอเพนซอร์ส (Open-Source) ที่พัฒนาและสนับสนุนโดย Google ภายใต้การนำของนักพัฒนาที่มีชื่อเสียง ได้แก่ Robert Griesemer, Rob Pike, และ Ken Thompson Go ได้รับการออกแบบมาเพื่อแก้ไขปัญหที่พบในภาษาโปรแกรมที่มีอยู่เดิม โดยเฉพาะอย่างยิ่งปัญหาที่เกี่ยวข้องกับ การพัฒนาซอฟต์แวร์ในยุคสมัยใหม่ ซึ่งเน้นที่ระบบเครือข่ายขนาดใหญ่ (Network Services) ระบบคลาวด์ (Cloud Computing) และการประมวลผลแบบมัลติคอร์ (Multi-core Processing)

ทฤษฎีพื้นฐานในการพัฒนา Go คือการสร้างภาษาที่สามารถรวมจุดเด่นของภาษาอื่น ๆ เข้าไว้ด้วยกัน โดยมีเป้าหมายหลัก 3 ประการคือ: 1. ความง่ายในการเขียนและทำความเข้าใจ (Simplicity) 2. ประสิทธิภาพในการประมวลผลและคอมไพล์ (Efficiency) และ 3. การรองรับการทำงานพร้อมกัน (Concurrency) ในระดับพื้นฐาน

2.4.1 หลักการทางทฤษฎีที่สำคัญในการออกแบบภาษา Go

ภาษา Go ไม่ได้ถูกสร้างขึ้นจากทฤษฎีทางคณิตศาสตร์ที่ซับซ้อน แต่สร้างขึ้นจากปรัชญาการออกแบบ (Design Philosophy) ที่มุ่งเน้นการปฏิบัติ (Pragmatism) เพื่อให้การพัฒนาระบบมีความรวดเร็ว ปลอดภัย และสามารถปรับขนาดได้ (Scalability) หลักการเหล่านี้ถือเป็นทฤษฎีเบื้องหลังที่อธิบายว่า Go คืออะไร

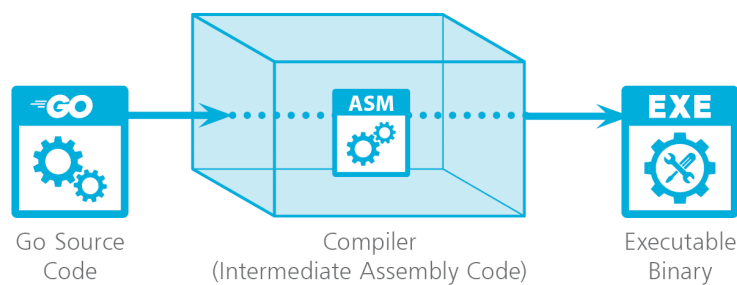
1. ทฤษฎีการทำงานพร้อมกันในระดับภาษา (First-Class Concurrency): หลักการที่สำคัญที่สุดของ Go คือการนำเสนอแนวคิดเรื่อง Concurrency (การทำงานพร้อมกัน) เข้ามาเป็นคุณสมบัติในระดับภาษาโดยตรง แทนที่จะพึ่งพาระบบปฏิบัติการหรือไลบรารีภายนอก ทฤษฎีนี้ประกอบด้วยสององค์ประกอบหลัก

- Goroutines: เป็นการทำงานพร้อมกันที่มีน้ำหนักเบามาก (Lightweight Thread) ซึ่งใช้หน่วยความจำน้อยกว่าเธรด (Thread) ทั่วไป ทำให้สามารถสร้าง Goroutines นับหมื่นหรือแสนรายการพร้อมกันได้อย่างมีประสิทธิภาพ นี่คือนโยบายที่สนับสนุนการสร้างระบบที่สามารถรองรับการเชื่อมต่อจำนวนมาก เช่น เซิร์ฟเวอร์เครือข่าย
- Channels: เป็นกลไกที่ใช้สำหรับ การสื่อสาร (Communication) ระหว่าง Goroutines เพื่อส่งผ่านข้อมูลและซิงโครไนซ์การทำงาน หลักการนี้สอดคล้องกับโมเดลการทำงานพร้อมกันที่เรียกว่า "Do not communicate by sharing memory; instead, share memory by communicating" ซึ่งช่วยลดปัญหาการเข้าถึงข้อมูลพร้อมกัน (Race Conditions) และเพิ่มความปลอดภัยของโค้ด

2. ทฤษฎีความเรียบง่ายและความชัดเจน (Simplicity and Clarity): Go ถูกออกแบบให้มีไวยากรณ์ (Syntax) ที่เรียบง่าย ดัดคุณสมบัติที่ซับซ้อนบางอย่างออกไป เช่น การสืบทอดแบบหลายคลาส (Multiple Inheritance) หรือตัวดำเนินการโอเวอร์โหลด (Operator Overloading) เพื่อให้โค้ด "มีรูปแบบเดียวที่ถูกต้อง (Idiomatic)" ทฤษฎีนี้เชื่อว่า

- โค้ดที่เรียบง่ายอ่านง่ายกว่า: ช่วยลดภาระทางความคิด (Cognitive Load) ของนักพัฒนาเมื่อต้องอ่านโค้ดของผู้อื่นในทีม

- การคอมไพล์ที่รวดเร็ว: ไวยากรณ์ที่ตรงไปตรงมาทำให้ตัวคอมไพเลอร์ (Compiler) สามารถทำงานได้อย่างรวดเร็ว ซึ่งส่งผลให้วงจรการเขียนโค้ด ทดสอบ และแก้ไข (Code, Test, Debug Cycles) สั้นลงอย่างมาก ซึ่งเป็นปัจจัยสำคัญในการเพิ่มประสิทธิภาพของทีมพัฒนา (Developer Productivity)
3. ทฤษฎีประสิทธิภาพสำหรับระบบคลาวด์และเครือข่าย (Efficiency for Cloud & Network Systems): Go ถูกออกแบบมาเพื่อเป็น ภาษาคอมไพล์ (Compiled Language) ที่แปลงเป็นไฟล์ปฏิบัติการแบบไบนารี (Single Static Binary) ทฤษฎีนี้มีข้อดีคือ
- การปรับใช้ที่ง่ายดาย (Easy Deployment): สามารถนำไปติดตั้งบนเซิร์ฟเวอร์หรือในคอนเทนเนอร์ (Container) เช่น Docker หรือ Kubernetes ได้โดยไม่ต้องติดตั้งไลบรารีเพิ่มเติม
 - ประสิทธิภาพการใช้หน่วยความจำ: Go มีระบบจัดการหน่วยความจำ (Garbage Collection) ที่มีประสิทธิภาพสูง ทำให้แอปพลิเคชันที่เขียนด้วย Go มักจะมี รอยเท้าหน่วยความจำที่เล็ก (Small Memory Footprints) และใช้ทรัพยากร CPU น้อยกว่าเมื่อเทียบกับภาษาอื่น ๆ ทำให้ประหยัดค่าใช้จ่ายในการดำเนินงานระบบคลาวด์



รูปที่ 2.4: กระบวนการทำงานของ Go ที่แปลงเป็น Executable

2.5 JSON

JSON (JavaScript Object Notation)[2] เป็นรูปแบบข้อมูลแบบข้อความ (text-based data format) ที่มีโครงสร้างเรียบง่าย อ่านเข้าใจได้ทั้งมนุษย์และคอมพิวเตอร์ และไม่ขึ้นอยู่กับภาษาโปรแกรมใดเป็นพิเศษ ออกแบบมาเพื่อใช้แลกเปลี่ยนข้อมูลผ่านเครือข่าย โดยเฉพาะในงานพัฒนาเว็บและบริการ API รูปแบบไวยากรณ์ของ JSON มีสองโครงสร้างหลัก ได้แก่

1. Object: ชุดของคู่ข้อมูล ชื่อ-ค่า (key-value pairs) ล้อมด้วยวงเล็บปีกกา ... โดยชื่อ (key) ต้องเป็น string และคั่นแต่ละคู่ด้วยเครื่องหมายจุลภาค ,
2. Array: ลิสต์ของค่าที่เรียงลำดับ ล้อมด้วยวงเล็บเหลี่ยม [...]

ชนิดข้อมูลพื้นฐานของ JSON ได้แก่ string, number, boolean, null, object, และ array โดยสตริงใช้การเข้ารหัสอักขระแบบ Unicode (ทั่วไปคือ UTF-8) และรองรับอักขระพิเศษด้วยการ escape ตัวเลขเป็นเลข

ทศนิยมฐานสิบโดยไม่ระบุชนิดย่อย (เช่น int หรือ float) ซึ่งขึ้นอยู่กับภาษาหรือ library ที่ใช้งานว่าจะ แปลงค่าดังกล่าวให้ตรงกับชนิดข้อมูลภายในของภาษาโปรแกรมนั้นอย่างไร คุณสมบัติเด่นของ JSON คือมีโครงสร้างที่ กะทัดรัด, เข้าใจง่าย, และเป็นมาตรฐานเดียวกัน ทำให้เหมาะกับการใช้งานใน REST/HTTP API, การกำหนดค่าระบบ (configuration), และการสื่อสารระหว่างบริการแบบ Microservices ในทางปฏิบัติ JSON มักถูกจัดรูป (pretty print) เพื่อให้อ่านง่ายระหว่างพัฒนา และบีบอัดเมื่อส่งข้อมูลผ่านเครือข่ายเพื่อประหยัด bandwidth นอกจากนี้ยังมีแนวทางการตรวจสอบความถูกต้องของข้อมูล เช่น JSON Schema ซึ่งช่วยยืนยันว่าโครงสร้างข้อมูลที่ได้รับหรือส่งตรงตามรูปแบบที่กำหนด ลดความผิดพลาดจากข้อมูลไม่ครบหรือชนิดไม่ตรง

2.6 Pandoc

Pandoc[5] เป็นเครื่องมือแปลงเอกสาร (document converter) แบบ open-source ที่พัฒนาโดย John MacFarlane โดยมีวัตถุประสงค์เพื่อใช้ในการแปลงไฟล์เอกสารจากรูปแบบหนึ่งไปสู่อีกรูปแบบหนึ่งได้อย่างยืดหยุ่นและมีความถูกต้องสูง ซึ่งได้รับความนิยมอย่างแพร่หลายทั้งในงานด้านเอกสาร งานวิชาการ และระบบจัดการเนื้อหาอัตโนมัติในซอฟต์แวร์ต่าง ๆ Pandoc ถูกพัฒนาโดยใช้ภาษา Haskell ซึ่งเหมาะสำหรับการประมวลผลข้อมูลเชิงโครงสร้างและงานที่ต้องการความถูกต้องของข้อมูลสูง โดยรองรับรูปแบบไฟล์มากกว่า 40 ประเภท เช่น DOCX, HTML, LaTeX, PDF, Markdown, EPUB และอื่น ๆ โดยผู้ใช้งานสามารถกำหนดรูปแบบอินพุต (input format) และเอาต์พุต (output format) ได้อย่างอิสระ ทั้งผ่าน Command Line Interface (CLI), ผ่าน Docker Container, หรือผ่านการเรียกใช้งานในรูปแบบ library ภาษา Haskell (Haskell API) ในทางเทคนิค Pandoc ทำงานผ่าน parser-writer pipeline โดยเริ่มจากการแปลงเอกสารต้นฉบับให้กลายเป็น Abstract Syntax Tree (AST) ภายใน จากนั้นจึงแปลง AST ดังกล่าวออกไปเป็นรูปแบบเอกสารเป้าหมาย ซึ่งช่วยให้สามารถรองรับโครงสร้างข้อมูลที่ซับซ้อน เช่น ตาราง (table), หมายเลขลำดับ (numbering), สมการ (equations) และสัญลักษณ์พิเศษต่าง ๆ ได้อย่างมีประสิทธิภาพ[4]

2.7 Regular expression

Regular expression (regex)[1] เป็นรูปแบบสัญลักษณ์ทางภาษาคอมพิวเตอร์ที่ใช้สำหรับอธิบายรูปแบบของข้อความ (text pattern) ซึ่งสามารถใช้ในการค้นหา ตรวจสอบ หรือดัดแปลงข้อความได้อย่างยืดหยุ่น Regular expression ถูกใช้อย่างแพร่หลายในงานประมวลผลข้อมูล การตรวจสอบความถูกต้องของข้อมูล (data validation), การกรองข้อความ (text filtering), และการพัฒนาโปรแกรมที่เกี่ยวข้องกับการประมวลผลสตริง แนวคิดของ regular expression มีรากฐานมาจาก ทฤษฎีภาษารูปแบบ (Formal Language Theory) และ ทฤษฎีออโตมาตา (Automata Theory) ซึ่งเริ่มต้นในช่วงทศวรรษ 1950 โดยนักคณิตศาสตร์ชื่อ Stephen Kleene ที่นิยาม "regular sets" เพื่ออธิบายรูปแบบของสัญลักษณ์ในภาษาเครื่องจักร ต่อมา มีการนำแนวคิดนี้ไปประยุกต์ใช้ในวิทยาการคอมพิวเตอร์เพื่อค้นหาข้อความในไฟล์และระบบต่าง ๆ ในเชิงโครงสร้าง Regular expression จะประกอบด้วย literal characters (ตัวอักษรธรรมดา) และ metacharacters (สัญลักษณ์พิเศษ) ซึ่งใช้ร่วมกันเพื่อกำหนดรูปแบบของข้อความ รูปแบบการเขียน regular expression มักใช้ syntax ที่เป็นภาษากลาง (pattern language) ซึ่งสามารถผสมอักขระธรรมดา กับสัญลักษณ์ควบคุมได้ เช่น

- `^A\d{3}$`: หมายถึงข้อความที่ขึ้นต้นด้วยตัวอักษร A และตามด้วยตัวเลข 3 ตัว และเป็นจุดสิ้นสุด

ของข้อความนั้น

- [A-Za-z0-9_]+: หมายถึงคำที่ประกอบด้วยตัวอักษร ตัวเลข หรือเครื่องหมาย underscore อย่างน้อยหนึ่งตัว

	Metacharacter	Metacharacter name	Meaning
1	^	caret	denote the beginning of a regular expression
2	\$	Dollar sign	denote the end of a regular expression or ending of a line
3	[]	Square bracket	check for any single character in the character set specified in []
4	()	Parenthesis	Check for a string. Create and store variables.
5	?	Question mark	check for zero or one occurrence of the preceding character
6	+	Plus sign	check for one or more occurrence of the preceding character
7	*	Multiply sign	check for any number of occurrences (including zero occurrences) of the preceding character.
8	.	Dot	check for a single character which is not the ending of a line
9		Pipe symbol	Logical OR
10	\	Escaping character	escape from the normal way a subsequent character is interpreted.
11	!	Exclamation symbol	Logical NOT
12	{ }	Curly Brackets	Repeat preceding character

รูปที่ 2.5: ตัวอย่างสัญลักษณ์ Metacharacter ที่ใช้ใน Regular Expression

บทที่ 3

โครงสร้างและขั้นตอนการทำงาน

ในบทนี้จะกล่าวถึงหลักการในการพัฒนาโครงงาน การออกแบบระบบ ของเว็บแอปพลิเคชัน และการออกแบบฐานข้อมูล

3.1 การสำรวจและทำความเข้าใจปัญหา

3.1.1 สัมภาษณ์และวิเคราะห์ผู้ใช้งาน

จากการเข้าพูดคุยถึงปัญหาที่เกิดขึ้นกับทางวิทยาลัยการศึกษาตลอดชีวิตในการลงทะเบียนหลักสูตรที่ผ่านมา พบว่าเกิดปัญหาหลักๆในการจัดการข้อมูลหลักสูตรที่มาจากหลากหลายคณะ และช่วงเวลาในการดำเนินการที่ไม่เท่ากัน และการติดตามสถานะของหลักสูตรที่เกิดปัญหาในการติดต่อสอบถามบ่อยครั้ง ทำให้เกิดความล่าช้าในการดำเนินการเปิดหลักสูตร และการเก็บข้อมูลหลักสูตรแบบส่วนตัวที่ไม่สามารถให้ผู้อื่นคนอื่นๆ เช่น เจ้าหน้าที่ของแต่ละคณะ หรือ อาจารย์ผู้เปิดหลักสูตร สามารถเข้ามาติดตามสถานะการดำเนินการของหลักสูตรได้ พร้อมกับปัญหาในการกรอกข้อมูลหลักสูตรลงในระบบเว็บไซต์ของวิทยาลัยการศึกษาตลอดชีวิตในการเปิดหลักสูตร โดยจะต้องคัดลอกข้อมูลจากไฟล์เอกสารที่ละบรรทัด ลงสู่เว็บไซต์ ทำให้เสียเวลาและเกิดความล่าช้าในการเปิดหลักสูตรได้ ซึ่งเป็นแนวคิดที่สามารถนำไปพัฒนาระบบเว็บแอปพลิเคชันเพื่อช่วยแก้ไขปัญหาดังกล่าวได้

3.1.2 สรุปปัญหาและตั้งวัตถุประสงค์ของโครงงาน

ดังนั้น เราจึงวางแผนในการแก้ไขปัญหาดังกล่าว ที่สามารถลดปัญหาเหล่านี้ได้ ในการสื่อสารกันระหว่าง เจ้าหน้าที่ของวิทยาลัยการศึกษาตลอดชีวิต และเจ้าหน้าที่ของคณะแต่ละคณะ โดยมีวัตถุประสงค์ที่จะเน้นในการเก็บหลักสูตรที่ต้องเปิดสอนและสามารถให้ผู้ใช้ติดตามสถานะของหลักสูตรได้ และการอำนวยความสะดวกในการเพิ่มข้อมูลหลักสูตรจากไฟล์เอกสารเข้าสู่ระบบได้ทันที เพื่อลดข้อผิดพลาดในการกรอกข้อมูลและล่าช้าในการดำเนินงาน โดยใช้เว็บแอปพลิเคชันเป็นสื่อกลางในการจัดการข้อมูลหลักสูตรทั้งหมด

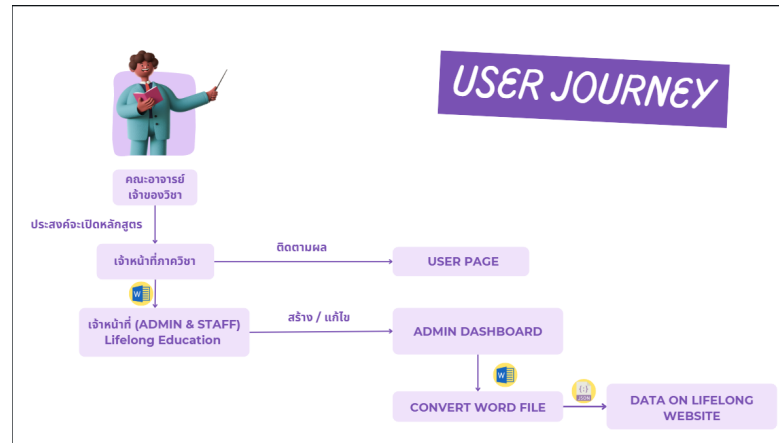
3.1.3 แผนภาพการดำเนินการส่งหลักสูตร

การจะดำเนินการส่งหลักสูตรนั้น จะต้องมีการสื่อสารกันระหว่าง เจ้าหน้าที่ของวิทยาลัยการศึกษาตลอดชีวิต และเจ้าหน้าที่ของคณะแต่ละคณะ และมีการดำเนินการตามขั้นตอนต่างๆภายในคณะ ถึงจะทำการส่งเอกสารหลักสูตรไปยังวิทยาลัยการศึกษาตลอดชีวิตได้ ดังแผนภาพต่อไปนี้

3.2 User interface (UI) and Design

3.2.1 User flow

หลังจากระบุความต้องการของผู้ใช้งานและวัตถุประสงค์ของโครงงานแล้ว จึงได้ทำการออกแบบ User flow เพื่อแสดงเส้นทางการใช้งานของผู้ใช้ในระบบเว็บแอปพลิเคชัน ว่าผู้ใช้จะต้องผ่านขั้นตอนใดบ้างในการใช้งานระบบ ตั้งแต่การเข้าสู่ระบบจนถึงการใช้งานฟังก์ชันต่างๆ ภายในระบบ และผู้ใช้สามารถดำเนินการอย่างไรได้บ้าง จึงออกแบบ User flow ดังแผนภาพต่อไปนี้



รูปที่ 3.1: User Flow ของระบบ

3.2.2 Role

เนื่องจากระบบเว็บแอปพลิเคชันนี้ มีผู้ใช้งานหลายประเภทที่มีบทบาทและสิทธิ์การเข้าถึงข้อมูลและฟังก์ชันการใช้งานที่แตกต่างกัน จึงได้กำหนดบทบาท (Role) ของผู้ใช้งานในระบบดังนี้

- **Admin** คือ ผู้ใช้ที่มีสิทธิ์สูงสุดในการจัดการระบบ สามารถเพิ่ม ลบ แก้ไข ข้อมูลผู้ใช้งาน และตั้งค่าระบบต่างๆ ได้
- **Staff** คือ ผู้ช่วย Admin ที่เป็นเจ้าหน้าที่ของวิทยาลัยการศึกษาลอตชีวิต ในการช่วยดำเนินการหลักสูตร
- **LE** คือ เจ้าหน้าที่ทั่วไปของวิทยาลัยการศึกษาลอตชีวิต ที่ต้องการจะตรวจสอบหลักสูตรที่ดำเนินการอยู่
- **Officer** คือ เจ้าหน้าที่ของคณะต่างๆ ที่มีหน้าที่ในการจัดการหลักสูตรภายในคณะและติดตามสถานะของหลักสูตรคณะของตน
- **User** คือ อาจารย์ผู้เปิดหลักสูตร ที่ต้องการส่งหลักสูตรไปยังวิทยาลัยการศึกษาลอตชีวิต

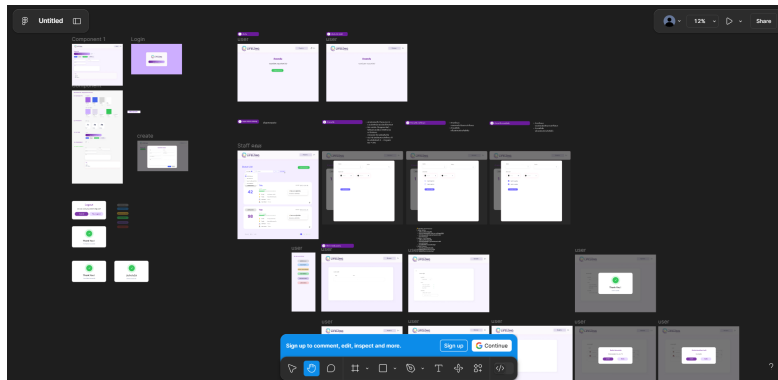
3.2.3 Design System

เมื่อกำหนด Flow การใช้งานและบทบาทของผู้ใช้งานในระบบแล้ว จึงได้ทำการออกแบบระบบดีไซน์ (Design System) เพื่อใช้เป็นแนวทางในการออกแบบส่วนติดต่อผู้ใช้งาน (User Interface) ของเว็บแอปพลิเคชัน ทั้งเรื่องการใช้ สี ฟอนต์ ไอคอน และองค์ประกอบต่างๆ ให้มีความสอดคล้องและเป็นมาตรฐานเดียวกันทั่วทั้งระบบตามวิทยาลัยการศึกษาลอตชีวิต โดยใช้ Tailwind CSS เป็นเครื่องมือในการพัฒนาระบบดีไซน์ ช่วยให้ผู้ใช้ สามารถสังเกตและองค์ประกอบต่างๆ ของระบบได้ง่ายขึ้น และเพิ่มประสิทธิภาพการใช้งานที่ดีให้กับผู้ใช้ ให้มีความเข้าใจง่าย และเป็นมิตรกับผู้ใช้งานมากที่สุด

3.2.4 User Interface Design

ผู้ใช้งานในระบบจะใช้งานผ่านเว็บไซต์ โดยใช้งานได้ทั้งระบบปฏิบัติการวินโดวส์ (Window) และ ไอโอเอส (IOS) ผ่านเว็บเบราว์เซอร์ สามารถใช้งานได้ทั้ง PC, Laptop และ Tablet โดยความเหมาะสมมาก

ที่สุด ควรใช้งานร่วมกับ PC หรือ Laptop เพื่อให้ได้ประสบการณ์ใช้งานที่ดีที่สุด โดยการออกแบบให้เห็นรายการของหลักสูตรเป็นการ์ด และเห็นลำดับและสถานะของหลักสูตรได้ชัดเจน รวมถึงปุ่มฟังก์ชันต่างๆ ที่ใช้งานบ่อยที่สุด ให้อยู่ในตำแหน่งที่ผู้ใช้สามารถเข้าใจและเข้าถึงได้ง่าย

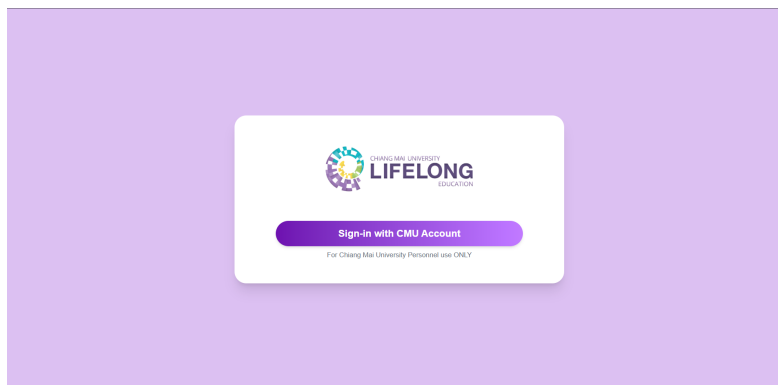


รูปที่ 3.2: figma ในการออกแบบหน้าตาระบบ

3.3 หลักการทำงานของเว็บแอปพลิเคชัน

3.3.1 หลักการทำงานของเว็บไซต์

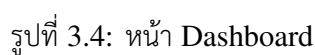
โดยผู้ใช้งานจะต้องทำการเข้าสู่ระบบผ่านทางหน้าเว็บไซต์ก่อน โดยใช้ CMU Account ในการยืนยันตัวตน ซึ่งสามารถเข้าได้เฉพาะผู้ที่มีสถานะเป็นเจ้าหน้าที่ของมหาวิทยาลัยเชียงใหม่ (MISEmpAcc) จากนั้นผู้ใช้งานสามารถเข้าถึงฟังก์ชันต่างๆ ภายในระบบได้ตามบทบาทของตนเอง



รูปที่ 3.3: หน้า Login

3.3.2 แดชบอร์ดหลัก

เป็นหน้าหลักที่ผู้ใช้งานเห็นหลังจากเข้าสู่ระบบสำเร็จ โดยจะแสดงภาพรวมของสถานะหลักสูตรทั้งหมดที่ผู้ใช้งานเกี่ยวข้อง เช่น หลักสูตรที่รอดำเนินการ หลักสูตรที่กำลังดำเนินการ และหลักสูตรที่เสร็จสิ้นแล้ว ผ่านการกรองสถานะของหลักสูตร โดยมีฟังก์ชันการใช้งานต่างๆ เช่น



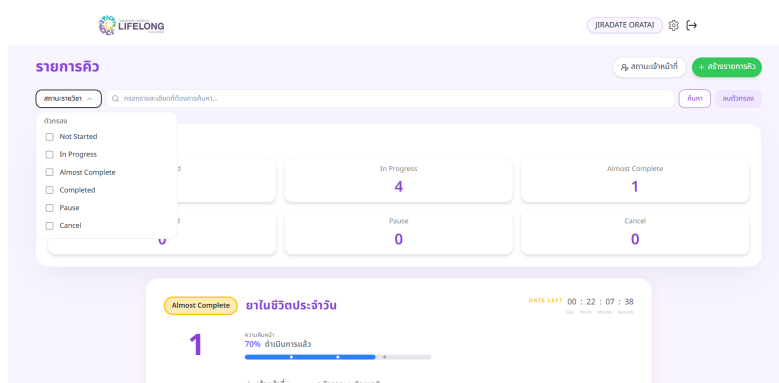
สามารถกรอกข้อมูลรายละเอียดของหลักสูตร เพื่อเป็นข้อมูลเบื้องต้นสำหรับติดตามข้อมูลการอัปเดตหลักสูตร เช่น ชื่อหลักสูตร คณะ เจ้าหน้าที่ผู้รับผิดชอบ วันที่ได้รับเอกสารไฟล์ Word หรือผู้รับผิดชอบหลักสูตร เป็นต้น

รูปที่ 3.5: หน้าสร้างหลักสูตรใหม่

สามารถแก้ไขข้อมูลรายละเอียดของหลักสูตรที่ได้สร้างไว้ก่อนหน้านี้ได้ และสามารถกรอกใช้รายการที่ต้องทำของแต่ละหลักสูตรได้ เพื่อแสดงความคืบหน้าของการดำเนินการของหลักสูตรนั้นๆ

สามารถรองสถานะของหลักสูตรได้ เช่น ยังไม่เริ่มต้น กำลังดำเนินการ เสร็จสิ้น หยุดการดำเนินการ และยกเลิก เพื่อให้ผู้ใช้สามารถดูรายการหลักสูตรตามสถานะที่ต้องการได้อย่างง่ายดาย

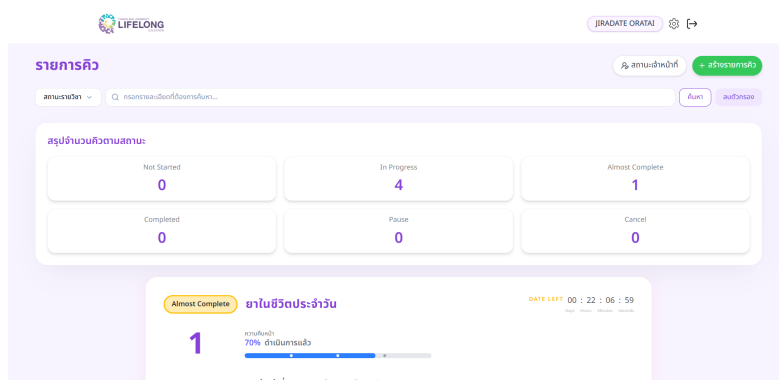
รูปที่ 3.6: หน้าแก้ไขหลักสูตรที่สร้าง



รูปที่ 3.7: หน้าฟังก์ชันในการกรอกรายการหลักสูตรด้วย course status

การค้นหาชื่อหลักสูตร

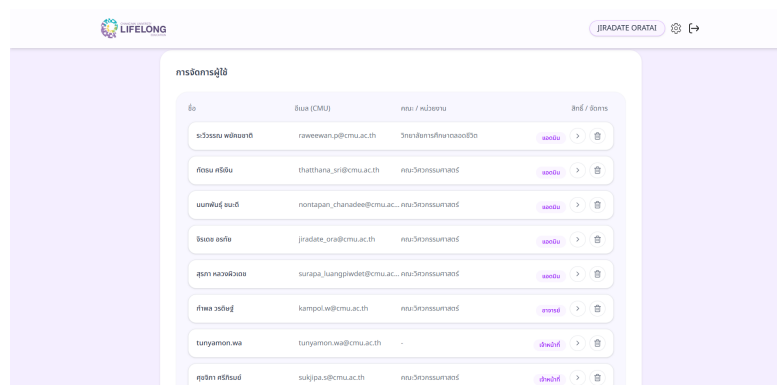
สามารถค้นหาหลักสูตรตามชื่อหลักสูตรได้อย่างรวดเร็ว โดยการพิมพ์คำค้นหาในช่องค้นหา เพื่อให้ผู้ใช้สามารถเข้าถึงข้อมูลหลักสูตรที่ต้องการได้ทันที



รูปที่ 3.8: หน้าฟังก์ชันค้นหารายการหลักสูตรด้วยชื่อหลักสูตร

การเพิ่มผู้ใช้ในระบบ

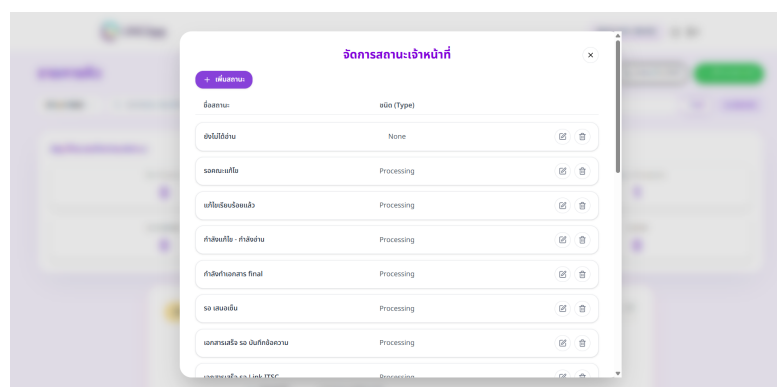
สามารถเพิ่มผู้ใช้ใหม่เข้าสู่ระบบได้ โดยกรอกข้อมูลรายละเอียดของผู้ใช้ คือ อีเมล CMU Account ของผู้ใช้ที่ต้องการจะเพิ่ม และบทบาทของผู้ใช้ เพื่อให้ผู้ใช้ใหม่สามารถเข้าสู่ระบบและใช้งานฟังก์ชันต่างๆ ได้ตามบทบาทที่กำหนด



รูปที่ 3.9: หน้าสำหรับจัดการผู้ใช้ในระบบ

การจัดการสถานะของเจ้าหน้าที่

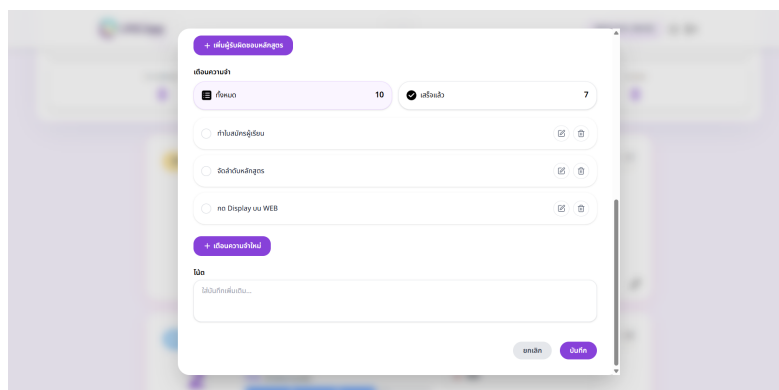
สามารถแก้ไขชื่อ ลบ และ เพิ่มสถานะของเจ้าหน้าที่ได้ เพื่อให้สามารถจัดการข้อมูลเจ้าหน้าที่ได้อย่างมีประสิทธิภาพ ในการแยกแยะรายการหลักสูตรตามประสงค์ของเจ้าหน้าที่



รูปที่ 3.10: หน้าสำหรับจัดการสถานะของเจ้าหน้าที่

การเพิ่มข้อมูลต่างๆ ที่จำเป็นต้องรายการหลักสูตร

สามารถเพิ่มข้อมูลต่างๆ ตามประสงค์ของเจ้าหน้าที่ได้ เช่น เพิ่มสถานะเจ้าหน้าที่ เพิ่มรายการที่ต้องทำของหลักสูตร เพิ่มผู้ใช้ในระบบ



รูปที่ 3.11: หน้าสำหรับจัดการรายการที่ต้องทำแต่ละหลักสูตร

3.4 การมองเห็นและการใช้งานแต่ละบทบาท

3.4.1 บทบาท Admin

เป็นผู้ดูแลระบบสูงสุด มีสิทธิ์ในการจัดการข้อมูลผู้ใช้ทั้งหมดในระบบ รวมถึงการตั้งค่าระบบต่างๆ เช่น การเพิ่ม ลบ แก้ไข ข้อมูลผู้ใช้งาน และตั้งค่าระบบต่างๆ ได้

3.4.2 บทบาท Staff

เป็นเจ้าหน้าที่ของวิทยาลัยการศึกษาตลอดชีวิต มีหน้าที่เป็นผู้ช่วย Admin ในการดำเนินการหลักสูตร โดยได้รับมอบหมายงานในการดูแลหลักสูตรจาก Admin สามารถใช้งานได้เหมือน Admin แต่ไม่สามารถเพิ่มผู้ใช้ใหม่เข้าสู่ระบบได้

3.4.3 บทบาท LE

เป็นเจ้าหน้าที่ของวิทยาลัยการศึกษาตลอดชีวิต มีหน้าที่ในสังเกตหลักสูตรที่กำลังดำเนินการอยู่ทั้งหมด แต่ไม่สามารถเพิ่ม ลด หรือ แก้ไขหลักสูตรได้ สามารถดูหลักสูตรทั้งหมดเท่านั้น

3.4.4 บทบาท Officer

เป็นเจ้าหน้าที่ของคณะต่างๆ มีหน้าที่ในการรายงานและติดตามสถานะหลักสูตรภายในคณะของตนเอง

3.4.5 บทบาท User

เป็นอาจารย์ผู้เปิดหลักสูตร มีหน้าที่ในการส่งหลักสูตรผ่านเจ้าหน้าที่ของคณะไปยังวิทยาลัยการศึกษาตลอดชีวิต และติดตามสถานะของหลักสูตรที่ตนเองได้ส่งไป

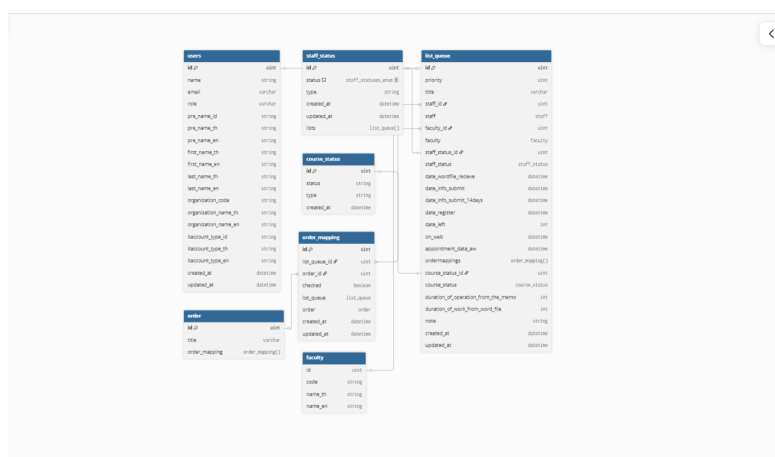
3.4.6 กรอกข้อมูลอัตโนมัติจากไฟล์เอกสาร

เป็นฟังก์ชันที่ช่วยอำนวยความสะดวกในการกรอกข้อมูลหลักสูตรจากไฟล์เอกสาร Word (.docx) โดยระบบจะทำการดึงข้อมูลที่จำเป็น เช่น ชื่อหลักสูตร, คณะ, รายละเอียดหลักสูตร เป็นต้น ในการกรอกข้อมูลเข้าสู่เว็บไซต์โดยอัตโนมัติ เพื่อลดข้อผิดพลาดในการกรอกข้อมูลและล่าช้าในการดำเนินงาน

3.5 การออกแบบฐานข้อมูล

3.5.1 โครงสร้างฐานข้อมูล

ฐานข้อมูลของระบบเว็บแอปพลิเคชันนี้ ถูกออกแบบให้มีความสัมพันธ์ระหว่างตารางต่างๆ อย่างเหมาะสม เพื่อรองรับการจัดเก็บข้อมูลหลักสูตร ผู้ใช้งาน สถานะของหลักสูตร คณะ และ รายการที่ต้องทำของแต่ละหลักสูตรได้อย่างมีประสิทธิภาพ โดยมีรายละเอียดของแต่ละตารางดังนี้



รูปที่ 3.12: ERdiagram สำหรับออกแบบระบบ

1. ตาราง list_queue: เก็บข้อมูลของหลักสูตรที่สร้างขึ้น โดยประกอบด้วยข้อมูลต่างๆ ดังนี้

- id: รหัสประจำหลักสูตร (Primary Key)
- priority: ลำดับของหลักสูตรในระบบ
- title: ชื่อหลักสูตร
- staff_id: รหัสผู้ดูแลหลักสูตร (Foreign Key)
- staff: ข้อมูลของ staff จาก staff_id นั้นๆ
- faculty_id: รหัสคณะ (Foreign Key)
- faculty: ข้อมูลของคณะจาก faculty_id นั้นๆ
- staff_status_id: รหัสสถานะของหลักสูตรตามประสงค์ของเจ้าหน้าที่ (Foreign Key)
- staff_status: ข้อมูลของสถานะจาก staff_status_id นั้นๆ
- date_wordfile_receive: วันที่ได้รับเอกสารไฟล์ Word
- date_into_submit: วันที่ส่งข้อมูลเข้าสู่ระบบ
- date_into_submit_14days: วันที่ส่งข้อมูลเข้าสู่ระบบภายใน 14 วัน
- date_register: วันที่ลงทะเบียนหลักสูตร
- date_left: วันที่เหลือในการเปิดหลักสูตร

- on_web: วันที่หลักสูตรต้องขึ้นเว็บไซต์ของวิทยาลัยการศึกษาตลอดชีวิต
 - appointment_data_aw: วันที่นัดหมายการประชุม
 - ordermapping: รายการที่ต้องทำในหลักสูตรนั้นๆ
 - course_status_id: รหัสสถานะของหลักสูตรตามประสงค์ของเจ้าหน้าที่
 - course_status: ข้อมูลของสถานะจาก course_status_id นั้นๆ
 - duration_of_operation_from_the_memo: ระยะเวลาการดำเนินการจากบันทึกข้อความ
 - duration_of_work_from_word_file: ระยะเวลาการดำเนินการจากไฟล์ Word
 - note: หมายเหตุเพิ่มเติม
 - created_at: วันที่และเวลาที่สร้างหลักสูตร
 - updated_at: วันที่และเวลาที่แก้ไขหลักสูตรล่าสุด
2. ตาราง staff_status: เก็บสถานะของหลักสูตรที่เจ้าหน้าที่กำหนดขึ้น โดยประกอบด้วยข้อมูลต่างๆ ดังนี้
- id: รหัสสถานะ (Primary Key)
 - status: ชื่อสถานะ
 - type: ชนิดของสถานะ
 - created_at: วันที่และเวลาที่สร้างสถานะ
 - updated_at: วันที่และเวลาที่แก้ไขสถานะล่าสุด
 - list: จำนวนรายการหลักสูตรที่มีสถานะนี้
3. ตาราง faculty: เก็บข้อมูลคณะต่างๆ ที่เกี่ยวข้องกับหลักสูตร โดยประกอบด้วยข้อมูลต่างๆ ดังนี้
- id: รหัสคณะ (Primary Key)
 - code: รหัสของคณะ
 - name_th: ชื่อคณะภาษาไทย
 - name_en: ชื่อคณะภาษาอังกฤษ
4. ตาราง course_status: เก็บข้อมูลรายชื่อสถานะของหลักสูตร
- id: รหัสสถานะ (Primary Key)
 - status: รายชื่อสถานะ
 - type: ชนิดของสถานะ
 - created_at: วันที่และเวลาที่สร้างสถานะ
5. ตาราง order: เก็บข้อมูลรายการที่ต้องทำของแต่ละหลักสูตร เพื่อแสดงความคืบหน้าของการดำเนินการ

- id: รหัสรายการ (Primary Key)
- title: รายชื่อรายการ
- order_mapping: ordermapping ที่มีรายการนี้ที่เกี่ยวข้อง

6. ตาราง order_mapping: เก็บข้อมูลรายการที่ต้องทำของแต่ละหลักสูตร ในการเชื่อมโยงระหว่างตาราง list_queue และ ตาราง order

- id: รหัสรายการจับคู่ (Primary Key)
- list_queue_id: รหัสรายการหลักสูตร (Foreign Key)
- order_id: รหัสรายการ (Foreign Key)
- checked: สถานะการทำรายการเสร็จสิ้นหรือไม่
- list_queue: ข้อมูลของหลักสูตรจาก list_queue_id นั้นๆ
- order: ข้อมูลของรายการจาก order_id นั้นๆ
- created_at: วันที่และเวลาที่สร้างรายการจับคู่
- updated_at: วันที่และเวลาที่แก้ไข

7. ตาราง users: เก็บข้อมูลผู้ใช้งานในระบบเว็บแอปพลิเคชัน

- id: รหัสผู้ใช้ (Primary Key)
- name: ชื่อผู้ใช้
- email: อีเมล CMU Account ของผู้ใช้
- role: บทบาทของผู้ใช้ในระบบ
- pre_name_id: คำนำหน้าชื่อ
- pre_name_th: คำนำหน้าชื่อภาษาไทย
- pre_name_en: คำนำหน้าชื่อภาษาอังกฤษ
- first_name_th: ชื่อภาษาไทย
- first_name_en: ชื่อภาษาอังกฤษ
- last_name_th: นามสกุลภาษาไทย
- last_name_en: นามสกุลภาษาอังกฤษ
- organization_code: รหัสของคณะ
- organization_name_th: ชื่อคณะภาษาไทย
- organization_name_en: ชื่อคณะภาษาอังกฤษ
- itaccount_type_id: รหัสประเภทผู้ใช้
- itaccount_type_th: รหัสประเภทผู้ใช้ภาษาไทย
- itaccount_type_en: รหัสประเภทผู้ใช้ภาษาอังกฤษ
- created_at: วันที่และเวลาที่สร้างผู้ใช้
- updated_at: วันที่และเวลาที่แก้ไขผู้ใช้ล่าสุด

3.6 API endpoint dashboard

- /swagger/*: สำหรับเข้าถึงเอกสารประกอบ API ผ่าน Swagger UI
- /health: Endpoint สำหรับตรวจสอบสถานะของ API (Health Check)
- /api/auth: Endpoint สำหรับการยืนยันตัวตนและการเข้าสู่ระบบ (Sign-In)

3.6.1 การตั้งค่า Middleware JWT

เส้นทางทั้งหมดที่อยู่ภายใต้ Group /api จะต้องผ่าน Middleware middleware.JWTProtected() เพื่อยืนยันตัวตนก่อนการเข้าถึงทรัพยากร

3.7 รายละเอียดการกำหนดเส้นทางตามบทบาท (Role-Based Routing)

ระบบมีการแบ่งกลุ่มเส้นทางออกเป็นสี่กลุ่มหลักตามบทบาทของผู้ใช้งาน (user, officer, LE, staff, และ admin)

3.7.1 Shared Routes (บทบาท: admin, staff, LE, officer, user)

เส้นทางเหล่านี้สามารถเข้าถึงได้โดยผู้ใช้งานทุกระดับที่เกี่ยวข้องกับระบบ โดยเน้นการดึงข้อมูลสถานะหลักสูตรและข้อมูลผู้ใช้งานปัจจุบัน

- GET /api/course/status: ดึงข้อมูลสถานะหลักสูตรทั้งหมด
- POST /api/listqueue/owner: ดึงรายการ listqueue ที่ผู้ใช้งานปัจจุบันเป็นเจ้าของหลักสูตร
- GET /api/user/me: ดึงข้อมูลผู้ใช้งานจากฐานข้อมูลสำหรับผู้ที่ใช้ระบบขณะนั้น

3.7.2 Officer Routes (บทบาท: admin, staff, LE, officer)

เส้นทางสำหรับเจ้าหน้าที่คณะ โดยเน้นการจัดการ listqueue ภายในคณะที่รับผิดชอบ

- POST /api/listqueue/faculty: ดึงรายการ listqueue ตามคณะของผู้ใช้งาน

3.7.3 LE Routes (บทบาท: admin, staff, LE)

เส้นทางสำหรับเจ้าหน้าที่วิทยาลัยการศึกษาตลอดชีวิต (Life-Long Education) เพื่อตรวจสอบภาพรวมของหลักสูตรทั้งหมด

- GET /api/listqueue: ดึงรายการ listqueue ทั้งหมด (รวมที่เสร็จสิ้นแล้ว)
- GET /api/listqueue/status/notyet: ดึงรายการ listqueue ที่ยังไม่เสร็จสิ้น
- POST /api/listqueue/staffstatus: ดึงรายการ listqueue ตามสถานะที่เจ้าหน้าที่กำหนด
- POST /api/listqueue/coursestatus: ดึงรายการ listqueue ตามสถานะหลักสูตร

3.7.4 Staff Routes (บทบาท: admin, staff)

เส้นทางหลักสำหรับการจัดการข้อมูลหลักสูตร การกำหนดลำดับ และรายการที่ต้องทำ ซึ่งจำกัดไว้เฉพาะเจ้าหน้าที่ (Staff) และผู้ดูแลระบบ (Admin)

การจัดการ ListQueue

- POST /api/listqueue: สร้างรายการ listqueue ใหม่
- PUT /api/listqueue: อัปเดตข้อมูลรายการ listqueue
- DELETE /api/listqueue/:id: ลบรายการ listqueue (จำกัดเฉพาะการพัฒนา)
- PUT /api/listqueue/:id/status/:staff_status_id: อัปเดตสถานะของ listqueue
- PUT /api/listqueue/:id/priority/:priority: อัปเดตลำดับความสำคัญของ listqueue

การจัดการ Order และ Staff Status

- GET /api/order/:id: ดึงรายการที่ต้องทำ (order) ของ listqueue นั้นๆ
- PUT /api/order: อัปเดตสถานะของรายการที่ต้องทำ (order)
- GET /api/staffstatus: ดึงข้อมูลสถานะที่เจ้าหน้าที่กำหนดทั้งหมด

การดึงข้อมูลประกอบ

- GET /api/staff: ดึงข้อมูลเจ้าหน้าที่ทั้งหมดสำหรับกำหนดผู้ดูแลหลักสูตร
- GET /api/faculty: ดึงข้อมูลคณะทั้งหมด

3.7.5 Admin Routes (บทบาท: admin)

เส้นทางสำหรับผู้ดูแลระบบ (Admin) เพื่อจัดการผู้ใช้งาน, รายการที่ต้องทำ, และสถานะหลักสูตร

การจัดการผู้ใช้งาน (Users)

- GET /api/user/all: ดึงข้อมูลผู้ใช้งานทุกคนในระบบ
- POST /api/user/:email/:role: สร้างผู้ใช้งานใหม่ พร้อมกำหนดบทบาท (role)
- DELETE /api/user/:id: ลบผู้ใช้งาน
- PUT /api/user/updateinfo/:email: อัปเดตข้อมูลผู้ใช้งาน
- GET /api/role: ดึงข้อมูลบทบาท (role) ทั้งหมด

การจัดการ Order

- POST /api/order: สร้างรายการ order ใหม่
- PUT /api/order/name: อัปเดตชื่อรายการ order
- DELETE /api/order/:id: ลบรายการ order

การจัดการ Staff Status

- POST /api/staffstatus: สร้างสถานะที่เจ้าหน้าที่กำหนดใหม่
- PUT /api/staffstatus/name: อัปเดตชื่อสถานะที่เจ้าหน้าที่กำหนด
- DELETE /api/staffstatus/:id: ลบสถานะที่เจ้าหน้าที่กำหนด

3.8 Course Document Converter

ระบบ Course Document Converter ถูกพัฒนาขึ้นเพื่อช่วยลดภาระงานซ้ำซ้อนของเจ้าหน้าที่ในวิทยาลัย การศึกษาตลอดชีวิต มหาวิทยาลัยเชียงใหม่ โดยเฉพาะในขั้นตอนการกรอกข้อมูลหลักสูตรลงในเว็บไซต์ของหน่วยงาน ระบบนี้มีหน้าที่หลักคือการอ่านไฟล์เอกสารหลักสูตรที่อยู่ในรูปแบบ .docx ซึ่งผ่านการจัดรูปแบบ ตามมาตรฐานแล้ว และทำการแปลงข้อมูลภายในให้เป็นโครงสร้างข้อมูลแบบ JSON เพื่อให้สามารถนำไป เติมนลงในแบบฟอร์มของเว็บไซต์ได้โดยอัตโนมัติ ทั้งนี้ระบบถูกออกแบบมาเพื่อทดแทนการคัดลอกและวางข้อความด้วยมือ ซึ่งมักใช้เวลานานและอาจเกิดข้อผิดพลาดจากมนุษย์ได้ง่าย การทำงานโดยรวมของระบบเริ่ม จากเมื่อเจ้าหน้าที่ทำการอัปโหลดไฟล์ข้อมูลหลักสูตรที่สมบูรณ์แล้วในรูปแบบ .docx ผ่านหน้าเว็บไซต์ ระบบ ฝั่ง server จะประมวลผลไฟล์ดังกล่าวและคืนค่าข้อมูลในรูปแบบ JSON ที่จัดเรียงตามหัวข้อมาตรฐาน ได้แก่

1. ชื่อหลักสูตรภาษาไทย
2. ชื่อหลักสูตรภาษาอังกฤษ
3. หน่วยงานที่เปิดหลักสูตร
4. จำนวนรับสมัคร
5. กลุ่มเป้าหมาย
6. หลักการและเหตุผล
7. วัตถุประสงค์
8. โครงสร้างหรือเนื้อหาของหลักสูตร
9. การประเมินผลตลอดหลักสูตร
10. คำสำคัญสำหรับการสืบค้น
11. คำอธิบายหลักสูตรอย่างย่อ
12. วันเปิดรับสมัคร
13. วันปิดรับสมัคร

14. วันสิ้นสุดของการชำระค่าธรรมเนียม
15. ค่าธรรมเนียมการอบรม
16. ค่าบำรุงมหาวิทยาลัย
17. ผู้ประสานงานหลักสูตร ได้แก่ คำนำหน้าชื่อ ชื่อ นามสกุล ตำแหน่ง หน่วยงาน ที่อยู่ หมายเลขโทรศัพท์ อีเมล และเว็บไซต์
18. หมวดหมู่การเรียนรู้

จากนั้นเว็บไซต์ของเจ้าหน้าที่จะนำข้อมูลดังกล่าวไปกรอกในช่องต่าง ๆ ของแบบฟอร์มโดยอัตโนมัติ เจ้าหน้าที่เพียงตรวจสอบความถูกต้องและกดยืนยันเท่านั้น ระบบประมวลผลเอกสารโดยมีขั้นตอนหลักดังนี้

1. ตรวจสอบและบันทึกไฟล์ที่อัปโหลดเมื่อผู้ใช้ส่งไฟล์ .docx เข้ามา ระบบจะตรวจสอบชนิดไฟล์และขนาดเพื่อให้แน่ใจว่าเป็นไฟล์ที่รองรับและไม่เกินขนาดสูงสุดที่กำหนดไว้ (10 MB) หากไม่ผ่าน ระบบจะส่งข้อความแสดงข้อผิดพลาดกลับไปพร้อมรหัสสถานะที่เหมาะสม
2. สร้างไฟล์ชั่วคราวสำหรับประมวลผลหลังจากตรวจสอบไฟล์ผ่านแล้ว ระบบจะสร้างไฟล์ชั่วคราวสามไฟล์ ได้แก่
 - input-*.docx สำหรับเก็บไฟล์ต้นฉบับที่อัปโหลด
 - output-*.txt สำหรับไฟล์ที่ถูกแปลงจากต้นฉบับเป็นข้อความธรรมดา (plain text)
 - output-*.html สำหรับไฟล์ที่ถูกแปลงจากต้นฉบับเป็น HTML
3. แปลงเอกสารด้วยโปรแกรม Pandoc ระบบเรียกโปรแกรม Pandoc สองครั้ง ครั้งแรกเพื่อแปลงจาก .docx เป็น plain text และครั้งที่สองเพื่อแปลงเป็น HTML การแปลงทั้งสองรูปแบบช่วยให้ระบบสามารถอ่านได้ทั้งข้อมูลเชิงข้อความและข้อมูลเชิงโครงสร้าง เช่น ตารางหรือรายการลำดับ ซึ่ง Pandoc มีความสามารถสูงในการรักษารูปแบบของเอกสารต้นฉบับไว้ได้อย่างถูกต้อง
4. แยกข้อมูลแต่ละหัวข้อจากเอกสาร ระบบจะนำไฟล์ผลลัพธ์จากขั้นตอนก่อนหน้าเข้ามาอ่านเนื้อหาที่ละบรรทัดจากไฟล์ plain text เพื่อดึงข้อมูลในแต่ละส่วนตามหัวข้อที่กำหนด และอ่านไฟล์ HTML เพื่อดึงข้อมูลที่อยู่ในรูปแบบตารางมาประกอบ ผลลัพธ์ทั้งหมดจะถูกรวมและจัดโครงสร้างให้อยู่ในรูปแบบ JSON
5. ส่งผลลัพธ์กลับไปยังผู้ใช้ เมื่อการประมวลผลเสร็จสมบูรณ์ ระบบจะส่งข้อมูล JSON ที่ได้กลับไปยังฝั่งผู้ใช้งาน เพื่อให้หน้าเว็บไซต์นำไปแสดงผลหรือเติมลงในช่องกรอกข้อมูลต่าง ๆ จากนั้นระบบจะทำการลบไฟล์ชั่วคราวทั้งหมดออกจากเครื่อง server โดยอัตโนมัติ
6. การจัดการข้อผิดพลาดและการบันทึก log ระบบมีการจัดการข้อผิดพลาดโดยจะบันทึกข้อความ log ทั้งในระดับข้อมูล (INFO) และข้อผิดพลาด (ERROR) พร้อมระบุรายละเอียดสาเหตุและไฟล์ที่เกี่ยวข้อง หากเกิดข้อผิดพลาดในขั้นตอนใด ระบบจะคืนรหัสสถานะ HTTP ที่เหมาะสม เช่น 413 สำหรับไฟล์ใหญ่เกินกำหนด หรือ 500 สำหรับข้อผิดพลาดภายใน เช่น การแปลงหรือการอ่านข้อมูลล้มเหลว

3.9 API endpoint document converter

สำหรับ document converter ด้วย course id

- POST /docx-converter-api/convert/:id: ใช้สำหรับอัปโหลดและแปลงไฟล์ .docx เป็น JSON เพื่อเติมข้อมูลหลักสูตรอัตโนมัติ

บทที่ 4

การทดลองและผลลัพธ์

4.1 การทดสอบการทำงานของระบบ

4.1.1 วัตถุประสงค์ในการทดสอบ

การทดสอบระบบมีวัตถุประสงค์เพื่อประเมินความถูกต้องของฟังก์ชันหลัก ความเสถียรของระบบ และความเหมาะสมของการออกแบบส่วนติดต่อผู้ใช้ (User Interface / User Experience) ภายใต้สภาวะการใช้งานจริง เพื่อยืนยันความพร้อมก่อนนำไปใช้งานภายในหน่วยงาน โดยมีวัตถุประสงค์ย่อยดังนี้

1. เพื่อทดสอบความถูกต้องของฟังก์ชันการทำงานให้เป็นไปตามที่ออกแบบ
2. เพื่อประเมินความถูกต้องของข้อมูลที่ประมวลผลและจัดเก็บในฐานข้อมูล
3. เพื่อประเมินความเหมาะสมของส่วนติดต่อผู้ใช้ (UI/UX) และการใช้งานในอุปกรณ์ต่าง ๆ
4. เพื่อทดสอบความพร้อมของระบบก่อนนำไปใช้งานจริงภายในหน่วยงาน

4.1.2 สภาพแวดล้อมของการทดสอบ (System Environment)

ระบบ CMU Lifelong Education Assistance Website ได้รับการออกแบบให้ทำงานบนสถาปัตยกรรมแบบ Web-Based Application ซึ่งสามารถเข้าถึงได้จากทุกแพลตฟอร์มโดยไม่ต้องติดตั้งโปรแกรมเพิ่มเติม มีรายละเอียดทางเทคนิคดังนี้

1. ข้อกำหนดด้านซอฟต์แวร์ (Software Requirements)
 - ใช้งานผ่าน Web Browser มาตรฐาน เช่น Google Chrome, Microsoft Edge, Safari และ Mozilla Firefox
 - รองรับการเข้าสู่ระบบเฉพาะผู้ใช้ที่มีบัญชีของมหาวิทยาลัยเชียงใหม่ (CMU Account - MIS-EmpAcc)
 - ประมวลผลแบบออนไลน์ผ่านโครงสร้าง Next.js (Frontend) และ Go Fiber (Backend) บนฐานข้อมูล PostgreSQL
2. ข้อกำหนดด้านฮาร์ดแวร์ (Hardware Requirements)
 - รองรับการใช้งานบน คอมพิวเตอร์ (PC / Laptop), แท็บเล็ต (Tablet) และ สมาร์ทโฟน (Smartphone) รวมถึง iPad
 - หน้าจอที่มีความละเอียดขั้นต่ำ 1152 x 854 พิกเซล เพื่อให้ส่วนติดต่อผู้ใช้แสดงผลได้อย่างถูกต้อง
 - ระบบออกแบบให้เป็น Responsive Design ปรับการแสดงผลอัตโนมัติให้เหมาะสมกับขนาดหน้าจอของอุปกรณ์แต่ละประเภท

จากผลการทดสอบเชิงเทคนิค พบว่าระบบสามารถทำงานได้ถูกต้องครบถ้วนในทุกอุปกรณ์ที่รองรับ ไม่มีปัญหาในการประมวลผลหรือการแสดงผลของส่วนติดต่อผู้ใช้ และสามารถใช้งานได้อย่างต่อเนื่องบนทุกเบราว์เซอร์ที่ทดสอบ

4.1.3 ขั้นตอนและกระบวนการทดสอบ

การทดสอบดำเนินการโดยจำลองการใช้งานจริงของผู้ใช้ในหน่วยงาน ครอบคลุมทุกบทบาท (Role) ได้แก่ Admin, Staff, LE, Officer, และ User โดยทดสอบการทำงานของแต่ละส่วนที่สัมพันธ์กับบทบาทนั้น ๆ ได้แก่

- การเข้าสู่ระบบด้วยบัญชี CMU Account
- การสร้างและจัดการข้อมูลหลักสูตร
- การติดตามสถานะการพิจารณาหลักสูตรแบบเรียลไทม์
- การจัดการผู้ใช้งานและสิทธิการเข้าถึง
- การแปลงไฟล์เอกสาร .docx ด้วยโมดูล Course Document Converter

ผลการทดสอบพบว่าระบบสามารถทำงานได้ครบตามที่ออกแบบไว้ โดยเฉพาะฟังก์ชันการแปลงเอกสารซึ่งช่วยลดระยะเวลาในการกรอกข้อมูลลงได้กว่า 60 เมื่อเทียบกับกระบวนการเดิมที่ต้องกรอกข้อมูลด้วยตนเอง

4.2 การทดสอบการใช้งานจริงของผู้ใช้

4.2.1 การออกแบบการทดสอบ

เพื่อประเมินประสบการณ์การใช้งานจริงของผู้ใช้ (User Experience Evaluation) ได้จัดทำแบบสอบถามออนไลน์ผ่าน Google Forms โดยมีผู้เข้าร่วมการประเมินจำนวน 6 คน ซึ่งครอบคลุมทุกบทบาทในระบบ ได้แก่ Admin, Staff, LE, Officer และ User ผู้เข้าร่วมแต่ละคนได้ทดลองใช้งานระบบในขั้นตอนสำคัญ เช่น

- การเข้าสู่ระบบด้วยบัญชี CMU Account
- การสร้างหลักสูตรใหม่
- การตรวจสอบและติดตามสถานะหลักสูตร
- การใช้ฟังก์ชันแปลงไฟล์เอกสารอัตโนมัติ

การประเมินแบ่งออกเป็น 2 ส่วน ได้แก่

1. การประเมินความพึงพอใจ (ระดับคะแนน 1-5) เพื่อประเมินด้านประสิทธิภาพ ความสะดวก และประโยชน์ต่อการทำงาน
2. การประเมิน Net Promoter Score (NPS) (ระดับคะแนน 1-10) เพื่อวัดความเต็มใจในการแนะนำระบบให้ผู้อื่นใช้งานต่อ

4.2.2 ผลการประเมินความพึงพอใจของผู้ใช้งาน

ผลการประเมินแสดงให้เห็นว่าผู้ใช้งานมีความพึงพอใจโดยรวมในระดับ “มาก” ถึง “มากที่สุด” โดยเฉพาะในด้านความเข้าใจง่ายของระบบ ความสะดวกในการใช้งาน และการแสดงผลที่รองรับอุปกรณ์พกพาได้อย่างดี

ด้านที่ประเมิน	ค่าเฉลี่ย	ระดับความพึงพอใจ
ความสะดวกในการใช้งานระบบ	4.50	พึงพอใจมาก
ความเข้าใจของส่วนติดต่อผู้ใช้ (UI/UX)	4.67	พึงพอใจมาก
ความถูกต้องของข้อมูลและสถานะหลักสูตร	4.60	พึงพอใจมาก
ความเร็วในการประมวลผล	4.20	พึงพอใจ
ความเหมาะสมต่อการใช้งานบนอุปกรณ์พกพา	4.50	พึงพอใจมาก
ประโยชน์ต่อการปฏิบัติงานจริง	4.80	พึงพอใจที่สุด
ค่าเฉลี่ยรวม	4.56	พึงพอใจมาก

ตารางที่ 4.1: ผลการประเมินความพึงพอใจของผู้ใช้งาน

4.2.3 ผลการประเมิน Net Promoter Score (NPS)

ผลการคำนวณค่า NPS เท่ากับ +33.3 จัดอยู่ในระดับ “ดี” แสดงให้เห็นว่าผู้ใช้งานส่วนใหญ่มีแนวโน้มที่จะแนะนำระบบให้ผู้อื่นใช้งานต่อ ซึ่งสะท้อนถึงความเชื่อมั่นในคุณภาพ ความสะดวก และประโยชน์ของระบบต่อกระบวนการทำงาน

กลุ่มผู้ตอบแบบสอบถาม	ช่วงคะแนน	ร้อยละ
Promoters (9–10 คะแนน)	3	50.0%
Passives (7–8 คะแนน)	2	33.3%
Detractors (0–6 คะแนน)	1	16.7%

ตารางที่ 4.2: ผลการประเมิน Net Promoter Score (NPS)

NPS Calculation:

$$\text{NPS} = \text{ร้อยละ Promoters} - \text{ร้อยละ Detractors}$$

$$\text{NPS} = 50.0 - 16.7 = +33.3$$

4.2.4 การรวบรวมความคิดเห็นจากผู้ใช้งาน (Feedback)

1. สิ่งที่ผู้ใช้งานชื่นชอบ

- ระบบใช้งานง่ายและไม่ซับซ้อน
- ลดระยะเวลาในการจัดการเอกสารหลักสูตรได้อย่างมีประสิทธิภาพ
- Dashboard แสดงผลชัดเจนและเข้าใจง่าย
- รองรับการใช้งานได้ดีทั้งบนคอมพิวเตอร์และอุปกรณ์พกพา

2. สิ่งที่ผู้ใช้งานต้องการให้ปรับปรุง

- เพิ่มระบบแจ้งเตือนอัตโนมัติ (Email / LINE OA)
- เพิ่มฟังก์ชันส่งออกข้อมูลเป็นไฟล์ Excel / PDF

- เพิ่มช่องทางติดต่อเจ้าหน้าที่ผ่านหน้าเว็บไซต์
- รองรับการแนบไฟล์เพิ่มเติมในระบบ

3. สิ่งที่ผู้ใช้งานสงสัยและเสนอแนะทางพัฒนาเพิ่มเติม

- ระบบสามารถแปลงไฟล์ Word ที่มี Template ต่างรูปแบบได้หรือไม่
- การรองรับผู้ใช้งานหลายคนพร้อมกันในเวลาเดียวกัน
- ความเป็นไปได้ในการเชื่อมต่อกับฐานข้อมูลกลางของ CMU MIS

4.2.5 สรุปผลการทดลอง

จากการทดสอบทั้งในเชิงเทคนิคและการใช้งานจริง พบว่าระบบ CMU Lifelong Education Assistance Website สามารถทำงานได้ครบถ้วนตามวัตถุประสงค์ที่ตั้งไว้ มีความเสถียร ถูกต้อง และช่วยลดเวลาในการปฏิบัติงานของเจ้าหน้าที่ได้อย่างมีนัยสำคัญ ผู้ใช้งานให้ความพึงพอใจโดยรวมในระดับ “มาก” (ค่าเฉลี่ย 4.56 จาก 5) และมีค่า NPS เท่ากับ +33.3 ซึ่งจัดอยู่ในระดับ “ดี” แสดงให้เห็นว่าระบบมีความพร้อมสำหรับการใช้งานจริงในระดับหน่วยงาน และสามารถพัฒนาต่อยอดได้ในอนาคต

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผล

โครงการเรื่อง ”ระบบติดตามการพิจารณาหลักสูตร วิทยาลัยการศึกษาลดชีวิต มหาวิทยาลัยเชียงใหม่ (CMU Lifelong Education Assistance Website)” มีวัตถุประสงค์เพื่อพัฒนาเว็บแอปพลิเคชันที่ช่วยให้

เจ้าหน้าที่สามารถบริหารจัดการและติดตามสถานะการพิจารณาหลักสูตรได้อย่างเป็นระบบ ลดภาระงานเอกสาร เพิ่มความโปร่งใส และเพิ่มประสิทธิภาพในการดำเนินงาน

ระบบพัฒนาด้วยเทคโนโลยี Next.js (Frontend), Go Fiber (Backend) และ PostgreSQL (Database) ภายใต้อสถาปัตยกรรมแบบ Hexagonal Architecture เพื่อความยืดหยุ่นและง่ายต่อการบำรุงรักษาใน

อนาคต ระบบสามารถทำงานได้บนอุปกรณ์ทุกประเภท ได้แก่ PC, Laptop, Tablet และ Smartphone โดยมีส่วนติดต่อผู้ใช้แบบ Responsive Design ที่สามารถปรับการแสดงผลตามขนาดหน้าจอได้อย่างเหมาะสม ผลการทดสอบพบว่าผู้ใช้งานให้ความพึงพอใจในระดับ “มาก” (ค่าเฉลี่ย 4.56) และมีค่า NPS เท่ากับ +33.3 ซึ่งสะท้อนถึงความเชื่อมั่นและศักยภาพของระบบในการนำไปใช้งานจริงภายในหน่วยงาน

5.2 ปัญหาที่พบและแนวทางการแก้ไข

5.2.1 การแปลงไฟล์เอกสารบางรูปแบบไม่สมบูรณ์

ในกรณีที่เอกสาร Word มีโครงสร้างซับซ้อน เช่น ตารางซ้อน (Nested Table) หรือ Template ไม่เป็นมาตรฐาน ระบบอาจไม่สามารถประมวลผลข้อมูลได้ครบถ้วน

แนวทางแก้ไข: ปรับปรุงโมดูล Document Converter ให้รองรับการตรวจสอบโครงสร้างเอกสารที่หลากหลายยิ่งขึ้น

5.2.2 การขาดระบบแจ้งเตือนอัตโนมัติ (Notification System)

ปัจจุบันยังไม่มีแจ้งเตือนเมื่อมีการเปลี่ยนแปลงสถานะหลักสูตร

แนวทางแก้ไข: พัฒนา Notification Service ผ่าน Email หรือ LINE OA เพื่อแจ้งสถานะแบบเรียลไทม์

5.2.3 ข้อจำกัดด้านเวลาการเก็บข้อมูลทดสอบจริง

จำนวนผู้เข้าร่วมการประเมินยังมีจำกัด

แนวทางแก้ไข: เพิ่มจำนวนกลุ่มผู้ใช้งานจากหลายคณะในการทดสอบรอบต่อไปเพื่อให้ได้ข้อมูลที่ครอบคลุมมากขึ้น

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

ข้อเสนอแนะเพื่อพัฒนาโครงการนี้ต่อไป มีดังนี้

- พัฒนา ระบบแจ้งเตือนอัตโนมัติ (Automated Notification System) เพื่อแจ้งสถานะหลักสูตรทันทีที่มีการเปลี่ยนแปลง
- เพิ่มฟังก์ชัน ส่งออกและนำเข้าข้อมูล (Export / Import Data) เป็น Excel หรือ PDF เพื่อใช้ในการจัดทำรายงาน
- พัฒนา Dashboard รวมระดับคณะและมหาวิทยาลัย เพื่อให้ผู้บริหารเห็นภาพรวมของหลักสูตรทั้งหมด
- เพิ่ม ระบบสื่อสารภายใน (Internal Messaging) สำหรับติดต่อระหว่างเจ้าหน้าที่และอาจารย์
- เชื่อมต่อกับ ระบบฐานข้อมูลกลางของมหาวิทยาลัย (CMU MIS Integration) เพื่อให้ข้อมูลอัปเดตโดยอัตโนมัติ
- ปรับปรุง การแสดงผลบนอุปกรณ์พกพา ให้มีความสอดคล้องกับประสบการณ์ผู้ใช้งานยิ่งขึ้น

บรรณานุกรม

- [1] Wikipedia contributors. *Regular expression*. Wikimedia Foundation, Inc., 2025. Accessed on October 22, 2025.
- [2] Douglas Crockford. *Introducing JSON*. 2002. Accessed on October 2025.
- [3] Developer. *TypeScript 101*. March 2020. Accessed on October 22, 2025.
- [4] John MacFarlane (jgm). *Pandoc: Universal markup converter*. GitHub, 2024. Version 3.8.2.1, Accessed on October 2025.
- [5] John MacFarlane. *Pandoc a universal document converter*. John MacFarlane, 2025.
- [6] Mikelopster. รู้จักกับ *Code Architecture*. 2025. Chapter 7, GoAPI Essential Course, Accessed on October 22, 2025.
- [7] The Go Project. *The Go Programming Language*. 2025. Accessed on October 23, 2025.
- [8] Inc. Vercel. *Next.js Docs*. 2025. Latest Version 16.0.0, Accessed on October 22, 2025.
- [9] โกเมศ อัมพวัน. บทที่ 10 การออกแบบฐานข้อมูล. เอกสารประกอบการสอน (Lecture Note). เข้าถึงเมื่อ 22 ตุลาคม 2568.

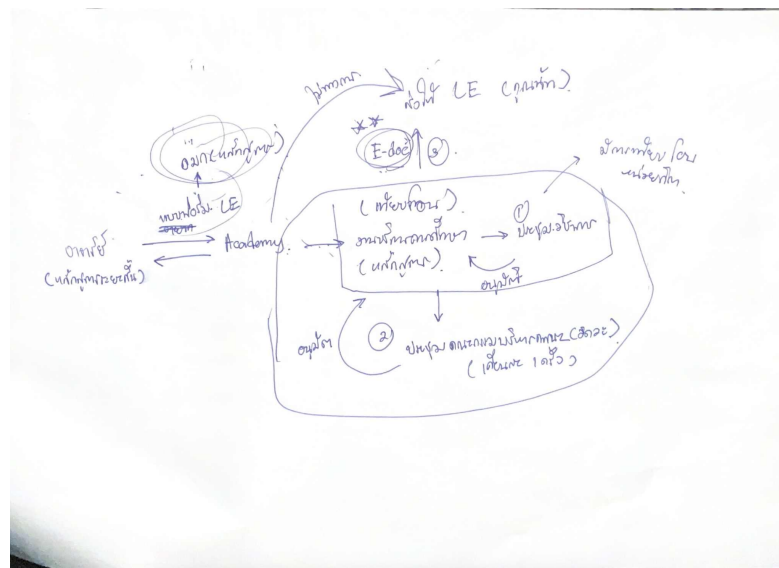
ภาคผนวก

การเก็บข้อมูลเพื่ออ้างอิงกับระบบ

ในส่วนนี้จะเป็นการเล่าถึงการพูดคุยรวมถึงการเก็บข้อมูลต่าง ๆ ที่เกี่ยวข้องกับระบบ ซึ่งส่งผลทำให้เกิดระบบนี้ขึ้นมา

ก.1 Flow การทำงานของ Entaneer Academy

จากการสำรวจข้อมูลเบื้องต้น พบว่า Entaneer Academy มีการทำงานในลักษณะดังภาพ ที่พบว่ามีปัญหา
มากมายในการส่งหลักสูตรให้กับวิทยาลัยการศึกษาดูแลชีวิต และในคณะอื่นๆก็มีปัญหาในระบบภายใน
ราชการเช่นเดียวกัน ส่งผลให้ระบบที่ถูกจำกัดคุณสมบัติบางอย่างไม่สามารถตอบโจทย์การทำงานได้ทั้งหมด
ที่ควรจะเป็นไปได้



รูปที่ ก.1: กระบวนการทำงานของ Entaneer Academy ในการส่งหลักสูตร

ก.2 Flow การทำงานของ วิทยาลัยการศึกษาดลัดชีวิต

จากการพูดคุยกับ stakeholder พบว่า วิทยาลัยการศึกษาลอตซีวิต จำเป็นต้องมีการขอหนังสือทางการ (e-doc) ในการยืนยันการส่งหลักสูตรในการดำเนินการเปิดหลักสูตร จึงทำให้มีขั้นตอนการทำงานที่ซับซ้อน เนื่องจากจะต้องทำงานร่วมกับหลายคณะที่มีการจัดการที่ไม่เหมือนกัน

ก.3 การได้รับ feedback หลักจากให้ผู้ใช้งานทดลองใช้ระบบ

จากการพูดคุยและนำเดโมระบบให้กับผู้ใช้งานจริง พบว่ามี **feedback** หลายๆส่วน เช่น การปรับเปลี่ยนจากระบบคิวเป็นระบบการรันเลขของหลักสูตร หรือ การเพิ่มบทบาทของผู้ใช้ในระบบ เป็นต้น ทำให้ทางผู้พัฒนาปรับเปลี่ยนระบบบ่อยครั้งให้สามารถ รองรับกับความต้องการของผู้ใช้งานได้มากที่สุด ซึ่งขอคำแนะนำจาก

ผศ.ดร. กำพล วรรดิษฐ์ ในการช่วยเหลือให้สามารถแก้ไขปัญหาได้ในที่สุด นำไปสู่การปรับปรุงระบบทั้งหมด และกำหนดแนวทางของการพัฒนาจนสำเร็จลุล่วง

ภาคผนวก ข

คู่มือการใช้งานระบบ

ข.1 Login

นักศึกษาที่ต้องการใช้เว็บไซต์ จำเป็นต้อง Login ผ่าน CMU OAuth โดยจะต้องเป็นบุคลากรของมหาวิทยาลัยเชียงใหม่ (MISEmpAcc)

ข.2 สร้างรายการหลักสูตรใหม่

เมื่อเข้าสู่ระบบสำเร็จ ให้เลือกเมนู "สร้างรายการหลักสูตรใหม่" จากนั้นกรอกข้อมูลหลักสูตรตามแบบฟอร์มที่กำหนด ซึ่งสามารถใช้งานได้เฉพาะบทบาท Admin และ Staff

ข.3 การแก้ไขหลักสูตร

เมื่อสร้างรายการหลักสูตรใหม่เสร็จสิ้นแล้ว เจ้าหน้าที่สามารถแก้ไขข้อมูลหลักสูตรได้จากมุมมองล่างของแต่ละรายการหลักสูตร

ข.4 เพิ่ม อัปเดต ลบ ผู้ใช้ในระบบ

เจ้าหน้าที่ที่มีบทบาท Admin สามารถจัดการผู้ใช้ในระบบได้ผ่านเมนู "ฟันเฟือง" โดยสามารถเพิ่มผู้ใช้ใหม่ อัปเดตข้อมูลผู้ใช้ หรือ ลบผู้ใช้ออกจากระบบได้ตามความเหมาะสม

ข.5 สร้าง ลบ เปลี่ยนชื่อ สถานะเจ้าหน้าที่

เจ้าหน้าที่สามารถสร้าง ลบ เปลี่ยนชื่อ สถานะเจ้าหน้าที่ได้ผ่านเมนู "สถานะเจ้าหน้าที่" ซึ่งสามารถเข้าถึงได้เฉพาะบทบาท Admin เท่านั้น

ข.6 สร้าง ลบ เปลี่ยนชื่อ รายการที่ต้องทำแต่ละรายการหลักสูตร

เจ้าหน้าที่สามารถสร้าง ลบ เปลี่ยนชื่อ รายการที่ต้องทำแต่ละรายการหลักสูตรได้ผ่านมุมมองล่างของแต่ละรายการหลักสูตร

ข.7 กรองสถานะ และการค้นหารายการหลักสูตร

ผู้ใช้งานสามารถกรองหลักสูตรได้ตามสถานะต่าง ๆ หรือ ค้นหารายการหลักสูตรด้วยคำสำคัญ (Keyword) ผ่านช่องค้นหาส่วนบนของหน้าเว็บไซต์

ประวัติผู้เขียน



ชื่อ-นามสกุล: นายจิรเดช อรทัย

รหัสนักศึกษา: 640610287

สาขาวิชา: ปริญญาตรี สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

Email : jiradate_ora@cmu.ac.th



ชื่อ-นามสกุล: นายนันทพันธุ์ ชนะดี

รหัสนักศึกษา: 640610645

สาขาวิชา: ปริญญาตรี สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

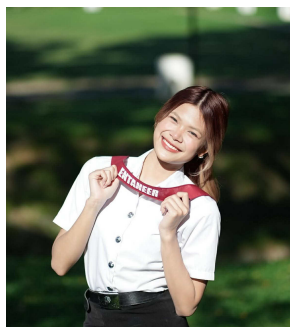
Email : nontapan_chanadee@cmu.ac.th



ชื่อ-นามสกุล: นายทัตธน ศรีเงิน

รหัสนักศึกษา: 640612088

สาขาวิชา: ปริญญาตรี สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
Email : thatthana_sri@cmu.ac.th



ชื่อ-นามสกุล: นางสาวสุรภา หลวงผิวเดช
รหัสนักศึกษา: 640612196
สาขาวิชา: ปริญญาตรี สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
Email : Surapa_luangpiwdet@cmu.ac.th