

โครงการเลขที่ วศ.คพ. P002-2/2567

เรื่อง

แพลตฟอร์มสำหรับจัดเก็บข้อมูลโปรเจกต์จบของวิศวกรรมศาสตร์ มหาวิทยาลัย  
เชียงใหม่

โดย

นายธนวินท์ สายทอง	รหัส 640610304
นายณฐพงศ์ พงศาวลีศรี	รหัส 640610630
นายพิชยุทธ หันชัยเนา	รหัส 640610653

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ปีการศึกษา 2567

**PROJECT No. CPE P002-2/2567**

**Project Box: A Platform for Archiving Senior Projects at CMU  
Engineering**

<b>Thanawin Saithong</b>	<b>640610304</b>
<b>Nathaphong Phongsawaleesri</b>	<b>640610630</b>
<b>Pichayut Hunchainao</b>	<b>640610653</b>

**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  
2024**

หัวข้อโครงการ : แพลตฟอร์มสำหรับจัดเก็บข้อมูลโปรเจกต์จบของวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่  
: Project Box: A Platform for Archiving Senior Projects at CMU Engineering

โดย : นายธนวินท์ สายทอง รหัส 640610304  
นายณฐพงศ์ พงศาสิทธิ์ รหัส 640610630  
นายพิชยุท หันชัยเนา รหัส 640610653

ภาควิชา : วิศวกรรมคอมพิวเตอร์  
อาจารย์ที่ปรึกษา : ผศ. โดม โพธิ์กานนท์  
ปริญญา : วิศวกรรมศาสตรบัณฑิต  
สาขา : วิศวกรรมคอมพิวเตอร์  
ปีการศึกษา : 2567

---

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์  
(รศ.ดร. สันติ พิทักษ์กัญญ์)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ  
(ผศ. โดม โพธิ์กานนท์)

..... กรรมการ  
(อ.ดร. ชินวัตร อิศราดิศัยกุล)

..... กรรมการ  
(นิรันดร์ พิสุทธอนนท์)

หัวข้อโครงการ : แพลตฟอร์มสำหรับจัดเก็บข้อมูลโปรเจกต์จบของวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่  
: Project Box: A Platform for Archiving Senior Projects at CMU Engineering

โดย : นายธนวินท์ สายทอง รหัส 640610304  
นายณฐพงศ์ พงศาวิศรี รหัส 640610630  
นายพิชยุท หันชัยเนา รหัส 640610653

ภาควิชา : วิศวกรรมคอมพิวเตอร์  
อาจารย์ที่ปรึกษา : ผศ. โดม โพธิ์กานนท์  
ปริญญา : วิศวกรรมศาสตรบัณฑิต  
สาขา : วิศวกรรมคอมพิวเตอร์  
ปีการศึกษา : 2567

---

### บทคัดย่อ

แพลตฟอร์มนี้ถูกพัฒนาขึ้นโดยมีจุดประสงค์หลักเพื่อเก็บรวบรวมโครงงานวิศวกรรม ของนักศึกษา คณะวิศวกรรมศาสตร์มหาวิทยาลัยเชียงใหม่ และใช้ในการสร้างแหล่งข้อมูลกลางที่เป็นระบบและเข้าถึงได้ง่าย นักศึกษาสามารถค้นหาและศึกษาโครงงานของรุ่นพี่ เพื่อนำไปใช้เป็นแรงบันดาลใจหรือตัวอย่างในการทำโครงงาน โดยระบบได้ออกแบบให้สามารถค้นหาได้จากหมวดหมู่ต่างๆ หรือสามารถค้นหาโครงงานที่เกี่ยวข้องได้โดยค้นหาจาก keyword ที่เกี่ยวข้องใน pdf ของโครงงานนั้นๆ

Project Title : Project Box: A Platform for Archiving Senior Projects at CMU Engineering  
Name : Thanawin Saithong 640610304  
Nathaphong Phongsawaleesri 640610630  
Pichayut Hunchainao 640610653  
Department : Computer Engineering  
Project Advisor : Asst.Prof.Dome Potikanond  
Degree : Bachelor of Engineering  
Program : Computer Engineering  
Academic Year : 2024

---

## **ABSTRACT**

This platform was developed with the main purpose of collecting engineering projects of students of the Faculty of Engineering, Chiang Mai University, and used to create a centralized and easily accessible source of information. Students can search for and study the projects of their seniors to use as inspiration or examples for doing projects. The system is designed to be searchable by various categories or to search for projects related to the room by searching for relevant keywords in the PDF of that project.

## กิตติกรรมประกาศ

โครงการนี้จะไม่สามารถสำเร็จได้ถ้าไม่ได้ความกรุณาจาก ผศ. โดม โพธิกานนท์ อาจารย์ที่ปรึกษาโครงการ ที่ได้สละเวลาให้ความช่วยเหลือให้คำแนะนำและสนับสนุนในการทำโครงการนี้รวมถึงอ.ดร.ชินวัตร อิศราดีสัยกุล และ ผศ.ดร.นิรันดร์ พิสุทธอานนท์ ที่ให้คำปรึกษาจนทำให้โครงการเล่มนี้เสร็จ สมบูรณ์ไปได้

นายธนวินท์ สายทอง  
นายณฐพงศ์ พงศ์วาลีศรี  
นายพิชญุต หันชัยเนาว์  
2 กุมภาพันธ์ 2567

## สารบัญ

บทคัดย่อ . . . . .	ข
Abstract . . . . .	ค
กิตติกรรมประกาศ . . . . .	ง
สารบัญ . . . . .	จ
สารบัญรูป . . . . .	ช
สารบัญตาราง . . . . .	ณ
<b>1 บทนำ . . . . .</b>	<b>1</b>
1.1 ที่มาของโครงการ . . . . .	1
1.2 วัตถุประสงค์ของโครงการ . . . . .	1
1.3 ขอบเขตของโครงการ . . . . .	1
1.3.1 ขอบเขตด้านฮาร์ดแวร์ . . . . .	1
1.3.2 ขอบเขตด้านซอฟต์แวร์ . . . . .	1
1.4 ประโยชน์ที่ได้รับ . . . . .	1
1.5 เทคโนโลยีและเครื่องมือที่ใช้ . . . . .	2
1.5.1 Next.js . . . . .	2
1.5.1.1 ข้อดีของ Next.js . . . . .	2
1.5.2 ElysiaJs . . . . .	2
1.5.3 Gin . . . . .	3
1.5.3.1 Feature สำคัญของ Gin . . . . .	3
1.5.4 RabbitMQ . . . . .	3
1.5.4.1 คำศัพท์ต่างๆที่เกี่ยวข้อง . . . . .	4
1.5.5 Docker . . . . .	4
1.5.6 องค์ประกอบหลักของ Docker . . . . .	5
1.5.6.1 ความแตกต่างระหว่าง Docker กับ Virtual Machines . . . . .	6
1.5.7 Nginx Proxy Manager (NPM) . . . . .	6
1.5.8 PostgreSQL . . . . .	7
1.5.9 Figma . . . . .	7
1.6 แผนการดำเนินงาน . . . . .	7
1.7 บทบาทและความรับผิดชอบ . . . . .	9
<b>2 ทฤษฎีที่เกี่ยวข้อง . . . . .</b>	<b>10</b>
2.1 ระบบฐานข้อมูล (Database System) . . . . .	10
2.2 ไมโครเซอร์วิส (Microservices) . . . . .	10
2.2.1 Monolithic VS Microservice . . . . .	10
2.2.1.1 ความแตกต่างระหว่าง Monolithic และ Microservice . . . . .	11
2.2.1.2 ข้อดีและข้อเสียของ Microservice . . . . .	11
2.3 Reverse Proxy . . . . .	12
2.3.1 ประโยชน์ของการทำ Reverse Proxy . . . . .	12
2.4 Hypertext Transfer Protocol (HTTP) . . . . .	12
2.5 Application Programming Interface (API) . . . . .	13
2.6 Json Web Token (JWT) . . . . .	13
2.7 Message Queue . . . . .	13
2.8 Model-View-Controller (MVC) . . . . .	14
2.9 Elasticsearch . . . . .	15

2.10 Inverted Index . . . . .	15
2.11 Kibana . . . . .	16
2.12 Spring Boot . . . . .	17
2.13 Hibernate . . . . .	17
<b>3 โครงสร้างและขั้นตอนการทำงาน</b>	<b>18</b>
3.1 สถาปัตยกรรมระบบ . . . . .	18
3.2 Frontend (Waiting for edit) . . . . .	18
3.3 Reverse Proxy . . . . .	19
3.4 Backend . . . . .	20
3.4.1 Auth Service . . . . .	20
3.4.1.1 Database Schema . . . . .	21
3.4.2 Project Service . . . . .	21
3.4.3 Search Service . . . . .	21
<b>4 การทดลองและผลลัพธ์</b>	<b>22</b>
4.1 การทดสอบฟังก์ชันการทำงาน . . . . .	22
4.1.1 การเข้าสู่ระบบของผู้ใช้ . . . . .	22
4.1.2 การค้นหาโครงการ . . . . .	22
4.1.2.1 การค้นหาแบบ Quick Search . . . . .	22
4.1.2.2 การค้นหาแบบ Advanced Search . . . . .	23
4.1.2.3 การค้นหาแบบ PDF Search . . . . .	23
4.1.2.4 การค้นหาแบบ Keyword Search . . . . .	23
4.1.3 การสร้างโครงการ . . . . .	24
<b>5 บทสรุปและข้อเสนอแนะ</b>	<b>25</b>
5.1 สรุปผล . . . . .	25
5.2 ปัญหาที่พบและแนวทางการแก้ไข . . . . .	25
5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ . . . . .	25
<b>บรรณานุกรม</b>	<b>26</b>
<b>ก คู่มือการติดตั้ง</b>	<b>27</b>
ก.1 Next.js 14 (Frontend) . . . . .	27
ก.1.1 Main Files and Folders . . . . .	27
ก.1.2 Subfolders . . . . .	28
ก.2 Installation Guide . . . . .	28
ก.2.1 Clone the Project from GitHub . . . . .	28
ก.2.2 Install Dependencies . . . . .	28
ก.2.3 Set Up Environment Variables . . . . .	28
ก.2.4 Run the Project in Development Mode . . . . .	28
ก.2.5 Build Docker Image (Optional) . . . . .	28
ก.3 Elysia (Auth_service) . . . . .	29
ก.3.1 Project Structure . . . . .	29
ก.3.2 Installation and Usage . . . . .	29
ก.4 Golang Project (Project_service) . . . . .	30
ก.4.1 Project Structure . . . . .	30
ก.4.2 Installation and Usage . . . . .	31



ก.5	Java Project (Search_service) . . . . .	32
ก.5.1	Project Structure . . . . .	32
ก.5.2	Installation and Usage . . . . .	32
ข	คู่มือการใช้งานระบบ	34

## สารบัญรูป

1.1	Next.js	2
1.2	Elysia	2
1.3	Gin	3
1.4	RabbitMQ	4
1.5	Docker	4
1.6	Nginx Proxy Manager	6
1.7	PostgreSQL	7
1.8	Figma	7
2.1	Monolithic Vs Microservice	11
2.2	Reverse Proxy	12
2.3	Message Queue	14
2.4	Model View Controller	14
2.5	Inverted Index	16
3.1	รูปภาพแสดงสถาปัตยกรรมของระบบ	18
3.2	รูปภาพแสดงการทแปลง IP ของ Nginx Reverse Proxy	19
3.3	รูปภาพแสดงการจัดการ SSL Certificate ของ Nginx Reverse Proxy	20
3.4	รูปภาพแสดง Database Schema ของ Auth Service	21
ข.1	ตัวอย่างไฟล์ Excel สำหรับนำเข้าข้อมูลนักศึกษา	40
ข.2	ตัวอย่างไฟล์ Excel สำหรับนำเข้าข้อมูลโครงการ	41
ข.3	รูปแบบการส่งโครงการ	42
ข.4	กำหนดสิ่งที่นักศึกษาต้องส่งในโครงการของแต่ละสาขา เช่น รายงาน, โค้ด, ฯลฯ	42
ข.6	เพิ่มข้อมูลอาจารย์ โดยการนำเข้าข้อมูลจากไฟล์ Excel ที่ได้จากสำนักทะเบียน	44
ข.5	ตัวอย่างไฟล์ Excel สำหรับนำเข้าข้อมูลอาจารย์	44
ข.7	ตัวอย่างไฟล์ Excel สำหรับนำเข้าข้อมูลอาจารย์	44
ข.8	จัดการข้อมูลสาขา	45
ข.9	เพิ่มเจ้าหน้าที่ประจำสาขา	46

## สารบัญตาราง

1.1	เปรียบเทียบคุณสมบัติของ Docker และ Virtual Machine (VM) . . . . .	6
-----	---	---

# บทที่ 1

## บทนำ

### 1.1 ที่มาของโครงการ

ที่มาของโครงการนั้นเกิดจากการเก็บโครงการในปัจจุบันขอภาควิศวกรรมศาสตร์ยังมีการเก็บการจัดกระจายผู้ที่สนใจต้องการเข้าถึงโครงการต้องติดต่อผู้ที่เก็บโครงการโดยตรง และไม่สามารถเข้าถึงโครงการได้เอง ทำให้เกิดปัญหาในการเข้าถึงข้อมูลที่ต้องการ

จากสาเหตุที่กล่าวมาข้างต้นทำให้ผู้พัฒนาต้องการพัฒนาแพลตฟอร์มที่เป็นศูนย์กลางการเก็บโครงการรวมถึงทำให้โครงการสามารถเข้าถึงได้โดยง่ายโดยการทำให้อยู่ในรูปแบบของ Web Application ที่ช่วยให้ผู้ใช้สามารถอัปโหลด จัดเก็บ และค้นหาโครงการได้อย่างเป็นระบบ

### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อสร้างระบบศูนย์กลางที่สามารถรวบรวมและจัดเก็บข้อมูลโปรเจกต์จบของนักศึกษาจากทุกภาควิชาในคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
2. ช่วยให้นักศึกษาสามารถค้นหาและศึกษาข้อมูลโปรเจกต์ รุ่นพี่เพื่อใช้อ้างอิงและเป็นแนวทาง
3. เพื่อง่ายต่อการค้นหาโปรเจกต์ที่อยู่นอกเหนือจากภาควิชาของตนเอง

### 1.3 ขอบเขตของโครงการ

#### 1.3.1 ขอบเขตด้านฮาร์ดแวร์

1. คอมพิวเตอร์หรือโน้ตบุ๊กที่ใช้สำหรับระบบต้องสามารถเชื่อมต่อสัญญาณอินเทอร์เน็ต

#### 1.3.2 ขอบเขตด้านซอฟต์แวร์

1. เว็บไซต์ไม่รองรับการแสดงผลแบบ Responsive ในขนาดหน้าจอที่เล็ก(เช่นใน โทรศัพท์มือถือ)

### 1.4 ประโยชน์ที่ได้รับ

1. ช่วยให้นักศึกษาและนักพัฒนาสามารถจัดเก็บและเข้าถึงโครงการได้ง่าย
2. เพิ่มโอกาสในการเรียนรู้จากโปรเจกต์ที่มีอยู่แล้ว
3. ช่วยองค์กรหรือมหาวิทยาลัยในการบริหารจัดการโปรเจกต์ของนักศึกษาได้อย่างมีประสิทธิภาพ

## 1.5 เทคโนโลยีและเครื่องมือที่ใช้

### 1.5.1 Next.js



รูปที่ 1.1: รูปจาก <https://medium.com/geekculture/why-should-you-learn-next-js-in-2021-what-are-the-benefits-8292d79bc50c>

Next.js[?] คือ JavaScript webapps framework ถูกสร้างขึ้น on top จาก library อย่าง React, Webpack, และ Babel ขึ้นมาอีกที มีจุดเด่นคือ เป็น SSR (server-side rendering) ตั้งแต่ต้น

#### 1.5.1.1 ข้อดีของ Next.js

- สามารถทำ SSR ได้ง่าย
- มีการจัดการ SEO ที่ดี
- Hot reload เวลาเราแก้ไขไฟล์ หน้าเว็บของเราจะถูก refresh โดยอัตโนมัติ
- Project Structure ที่ชัดเจนที่ถูกออกแบบมาให้เรียบง่ายแล้ว
- Routing ด้วยความที่มี project structure การทำ routing จึงสามารถ auto routing ได้

### 1.5.2 ElysiaJs



รูปที่ 1.2: รูปจาก <https://sadewawicak25.medium.com/file-upload-and-security-validation-on-elysia-js-2-d6c57b023441>

ElysiaJS[?] คือ Framework ในการพัฒนา API ด้วยภาษา Typescript โดยมีจุดเด่นคือ ความเร็วที่เร็วกว่า Express ถึง 21 เท่า (เนื่องจาก ElysiaJS มีการใช้ Bun เป็น Runtime) และอีกจุดเด่นคือ End-to-end Type Safety หรือชนิดของข้อมูลที่ชัดเจน ทำให้เวลาเราทำงานร่วมกับผู้อื่นสามารถทำได้สะดวกมากยิ่งขึ้น เพราะไม่ต้องมาทะเลาะกันเรื่องชนิดของข้อมูลที่ส่งให้กัน อีกทั้งยังมี Community ที่เติบโตเร็ว

### 1.5.3 Gin



รูปที่ 1.3: รูปจาก <https://www.askme.co.th/article/what-is-docker/>

Gin[?] เป็น web framework ที่เขียนด้วยภาษา go lang ที่ถูกพัฒนาต่อมาจาก Martini API ที่หยุดพัฒนาไปแล้ว โดย Gin จะใช้ customized httprouter ทำให้มีประสิทธิภาพด้านความเร็วที่สูงมากกว่า Martini ถึง 40 [?]เท่า ทำให้มีperformance กับ productivity ที่ดี

#### 1.5.3.1 Feature สำคัญของ Gin

- **JSON validation** สามารถแปลงและตรวจสอบ JSON ของ HTTP request
- **Routes grouping** จัดกลุ่ม routes ของ request ว่า request ใหนต้องมีการ authorization หรือไม่จำเป็นต้องมี การแยก request ด้วย version ของ API โดยสามารถจัดกลุ่มได้อย่างไม่จำกัด และไม่กระทบกับประสิทธิภาพ
- **Middleware support** incoming HTTP request จะถูกจัดการด้วย chain ของ middleware และ action สุดท้าย
- **Rendering build-in** ง่ายสำหรับสร้าง API ที่ render เป็น JSON, XML และ HTML
- **Error management** สามารถจัดการ error ที่เกิดขึ้นในระดับ application และ HTTP ได้

### 1.5.4 RabbitMQ

RabbitMQ [?]ซอฟต์แวร์ที่เป็นตัวกลางรับส่งข้อความระหว่างแอปพลิเคชันต่างๆ ผู้ไปรับรับข้อความจากผู้ส่ง (แอปพลิเคชันหนึ่ง) เก็บไว้รอการคัดแยก และส่งต่อให้ผู้รับ (แอปพลิเคชันอีกแอปพลิเคชันหนึ่ง) เหมาะสำหรับการทำแอปพลิเคชันที่ต้องมีการจัดคิวในการส่งข้อความ ระบบที่เป็นไมโครเซอร์วิสเอาไว้สื่อสารกันได้อย่างมีประสิทธิภาพ ทำให้สามารถแบ่งงานขนาดใหญ่เป็นงานย่อยๆ และส่งไปยังระบบอื่นๆ เพื่อประมวลผลได้นั่นเอง



รูปที่ 1.4: รูปจาก <https://www.borntodev.com/2024/06/09/rabbitmq-nodejs/>

#### 1.5.4.1 คำศัพท์ต่างๆที่เกี่ยวข้อง

- **Producer** คือ ผู้ส่งข้อความ
- **Consumer** คือ ผู้รับข้อความ
- **Queue** คือ คิวข้อความ
- **Exchange** คือ ตัวกลางในการส่งข้อความ
- **Binding** คือ การเชื่อมต่อระหว่าง Exchange กับ Queue
- **Channel** คือ ช่องสื่อสารระหว่าง Producer และ Consumer
- **Connection** คือ การเชื่อมต่อระหว่าง RabbitMQ กับ Producer และ Consumer

#### 1.5.5 Docker



รูปที่ 1.5: รูปจาก <https://www.docker.com/>

Docker[?] เป็นแพลตฟอร์มโอเพนซอร์สที่ช่วยในการสร้าง ทดสอบ และปรับใช้แอปพลิเคชันในรูปแบบของคอนเทนเนอร์ คอนเทนเนอร์เป็นสภาพแวดล้อมที่แยกจากกันที่สามารถรันแอปพลิเคชันได้ โดยไม่ต้องกังวลเกี่ยวกับการกำหนดค่าหรือการติดตั้งซอฟต์แวร์เพิ่มเติมในระบบปฏิบัติการหลักของเซิร์ฟเวอร์ที่ใช้

สิ่งที่ทำให้ Docker แตกต่างจากเทคโนโลยีอื่นๆ เช่น Virtual Machines (VMs) คือ Docker ใช้ Kernel ของระบบปฏิบัติการเดียวกันในการรันคอนเทนเนอร์แต่ละตัว ทำให้มีประสิทธิภาพในการใช้งานทรัพยากรสูงขึ้น และทำให้คอนเทนเนอร์ใช้เวลาในการเริ่มต้นที่รวดเร็ว

### 1.5.6 องค์ประกอบหลักของ Docker

- **Docker Engine** เป็นซอฟต์แวร์ที่รันอยู่เบื้องหลังซึ่งทำหน้าที่สร้างและจัดการคอนเทนเนอร์ในระบบ มันมีองค์ประกอบย่อย 2 ส่วนที่สำคัญ ได้แก่
  1. **Docker Daemon (dockerd)**: เป็นโปรแกรมหลักที่รันอยู่เบื้องหลังและรับคำสั่งจาก Docker Client ผ่าน API โดย Daemon จะจัดการทุกอย่างที่เกี่ยวข้องกับคอนเทนเนอร์ เช่น การสร้าง, การเริ่มต้น, การหยุด, และการลบคอนเทนเนอร์
  2. **Docker CLI (docker)**: คือส่วนที่ผู้ใช้ใช้ในการสั่งการ Docker ผ่านคำสั่งต่างๆ ผ่าน Command Line Interface เช่น การสร้างคอนเทนเนอร์ (docker run), การสร้างอิมเมจ (docker build), และการจัดการเครือข่าย (docker network)
- **Docker Image** คือไฟล์แบบคงที่ที่บรรจุโค้ดแอปพลิเคชันและทุกสิ่งที่แอปพลิเคชันนั้นต้องการในการรัน เช่น ไลบรารี, การตั้งค่า, และไฟล์ระบบ Image ถูกสร้างจากไฟล์ที่เรียกว่า Dockerfile ซึ่งเป็นไฟล์ที่กำหนดขั้นตอนในการติดตั้งและตั้งค่าแอปพลิเคชัน

**Dockerfile Syntax** ที่ใช้ในการสร้าง Image มีคำสั่งที่เป็นลำดับขั้นตอน ยกตัวอย่างเช่น

1. **FROM**: ระบุ Image เบื้องต้น เช่น Ubuntu, Alpine หรือ Node.js
  2. **RUN**: รันคำสั่งใน Image เช่น การติดตั้งแพ็คเกจ
  3. **COPY/ADD**: คัดลอกไฟล์จากโฮสต์เข้าสู่ Image
  4. **CMD/ENTRYPOINT**: กำหนดคำสั่งที่รันเมื่อคอนเทนเนอร์เริ่มทำงาน
- **Docker Container** คือ สิ่งที่ถูกสร้างจาก Docker Image และเป็นสภาพแวดล้อมที่แยกจากกันที่สามารถรันแอปพลิเคชันได้โดยมีคุณสมบัติดังนี้
    1. แยกการทำงานจากระบบปฏิบัติการโฮสต์ แต่ยังใช้เคอร์เนลร่วมกัน
    2. สามารถสร้าง, ลบ, หยุด, และรีสตาร์ทได้อย่างง่ายดาย
    3. สามารถแชร์ทรัพยากรเครือข่ายและไฟล์ระหว่างคอนเทนเนอร์ต่างๆ ได้



### 1.5.6.1 ความแตกต่างระหว่าง Docker กับ Virtual Machines

ตารางที่ 1.1: เปรียบเทียบคุณสมบัติของ Docker และ Virtual Machine (VM)

คุณสมบัติ	Docker	Virtual Machine (VM)
การใช้เคอร์เนล	ใช้เคอร์เนลของระบบปฏิบัติการโฮสต์	จำลองระบบปฏิบัติการเต็มรูปแบบ
ขนาดไฟล์	ขนาดเล็ก	ขนาดใหญ่
เวลาเริ่มต้น	เริ่มต้นได้อย่างรวดเร็ว	ใช้เวลามากขึ้น
การใช้ทรัพยากร	ใช้ทรัพยากรน้อย	ใช้ทรัพยากรมาก
ความยืดหยุ่น	เหมาะสำหรับแอปพลิเคชันแบบไมโครเซอร์วิส	เหมาะสำหรับการจำลองระบบขนาดใหญ่
การแยกทรัพยากร	แยกการทำงานในระดับแอปพลิเคชัน	แยกระบบปฏิบัติการทั้งหมด

โดยรวมแล้ว Docker เหมาะสำหรับการรันแอปพลิเคชันที่ต้องการความยืดหยุ่นในการปรับใช้และทรัพยากรที่มีข้อจำกัด ในขณะที่ VM เหมาะกับการรันระบบที่ต้องการการแยกสภาพแวดล้อมอย่างสมบูรณ์ เช่น การรันหลายระบบปฏิบัติการในเครื่องเดียวกัน

### 1.5.7 Nginx Proxy Manager (NPM)



รูปที่ 1.6: <https://sparwan.com/en/blogs/news/tutorial-installation-de-nginx-proxy-manager>

เป็น open-source ที่ออกแบบมาเพื่อช่วยให้การจัดการพร็อกซีของ Nginx, SSL, Access Lists และอื่นๆ โดยสร้างขึ้นมาเพื่อจุดประสงค์ให้ง่ายต่อการใช้งานโดยมี Dashboard ให้เหมาะสำหรับผู้ที่ไม่เชี่ยวชาญการใช้ Nginx ผ่าน CLI นอกจากนี้ ยังรองรับ SSL ฟรีผ่าน Let's Encrypt รวมถึงสามารถใช้งานได้บน Docker และรองรับการใช้งานหลาย User อีกด้วย

### 1.5.8 PostgreSQL



รูปที่ 1.7: รูปจาก <https://www.fullstackpython.com/postgresql.html>

PostgreSQL คือ ระบบฐานข้อมูลแบบ Relational Database Management System (RDBMS) ที่เป็นโอเพนซอร์ส มีความสามารถในการจัดการข้อมูลที่มีความซับซ้อน รวมถึงการจัดการข้อมูลที่มีความสัมพันธ์กัน

### 1.5.9 Figma



รูปที่ 1.8: รูปจาก <https://www.figma.com/>

Figma เป็นเครื่องมือออกแบบ UI/UX ที่ใช้งานผ่านเว็บ ทำให้สามารถทำงานร่วมกันได้ง่าย สามารถสร้าง Wireframe, Mockup, Prototype และ Design System ได้

## 1.6 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ต.ค. 2567	พ.ย. 2567	ธ.ค. 2567	ม.ค. 2568	ก.พ. 2568	มี.ค. 2568
ศึกษาค้นคว้า						
ออกแบบ Design UX/UI						
Implementation Frontend						
Deploy Services						
ทดสอบ						

ขั้นตอนการดำเนินงาน	ต.ค. 2567	พ.ย. 2567	ธ.ค. 2567	ม.ค. 2568	ก.พ. 2568	มี.ค. 2568
ศึกษาค้นคว้า						
ออกแบบ Database AuthService						
Implementation AuthService Backend						
Deploy Services						
ทดสอบ						

ขั้นตอนการดำเนินงาน	ต.ค. 2567	พ.ย. 2567	ธ.ค. 2567	ม.ค. 2568	ก.พ. 2568	มี.ค. 2568
ศึกษาค้นคว้า						
ออกแบบ Database ProjectService						
Implementation ProjectService Backend						
Deploy Services						
ทดสอบ						

ขั้นตอนการดำเนินงาน	ต.ค. 2567	พ.ย. 2567	ธ.ค. 2567	ม.ค. 2568	ก.พ. 2568	มี.ค. 2568
ศึกษาค้นคว้า						
ออกแบบ Database SearchService						
Implementation SearchService Backend						
Deploy Services						
ทดสอบ						

## 1.7 บทบาทและความรับผิดชอบ

- นายณฐพงศ์ พงศาสิทธิ์ รับผิดชอบในการพัฒนา Frontend , AuthService Backend , Deploy Services
- นายพิชยุทธ หันชัยเนาว์สุขเกษม รับผิดชอบในออกแบบ UX/UI รวมถึงการพัฒนา Frontend
- นายธนวินท์ สายทอง รับผิดชอบในการพัฒนา SearchService Backend , ProjectService Backend

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

การทำโครงการ เริ่มต้นด้วยการศึกษาค้นคว้า ทฤษฎีที่เกี่ยวข้อง หรือ งานวิจัย/โครงการ ที่เคยมีผู้นำเสนอไว้แล้ว ซึ่งเนื้อหาในบทนี้จะเกี่ยวกับการอธิบายถึงสิ่งที่เกี่ยวข้องกับโครงการ เพื่อให้ผู้อ่านเข้าใจเนื้อหาในบทถัดๆ ไปได้ง่ายขึ้น

#### 2.1 ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล (Database System) หมายถึงโครงสร้างสารสนเทศที่ประกอบด้วย รายละเอียดของข้อมูลที่เกี่ยวข้องกันที่จะนำมาใช้ในระบบต่าง ๆ ร่วมกัน ซึ่งผู้ใช้สามารถจัดการกับ ข้อมูลได้ในลักษณะต่าง ๆ ทั้งเพิ่ม ลบ หรือแก้ไขตลอดจนการเรียกดูข้อมูล ส่วนใหญ่จะเป็นการประยุกต์นำเอาระบบคอมพิวเตอร์เข้ามาช่วยในการจัดการฐานข้อมูล ระบบฐานข้อมูล มีคำศัพท์ต่างๆที่เกี่ยวข้องดังนี้

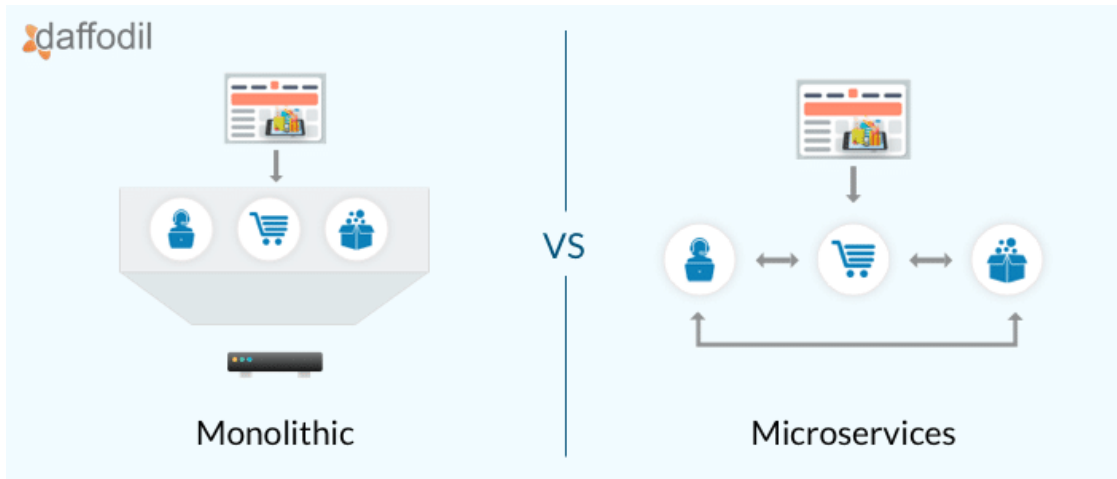
1. เอนทิตี (Entity) หมายถึงสิ่งที่เราสนใจจะเก็บข้อมูล เช่น นักศึกษา อาจารย์ วิชาการ หรือห้องเรียน
2. แอตทริบิวต์ (Attribute) หมายถึงคุณสมบัติของเอนทิตี เช่น ชื่อ นามสกุล หรือรหัสนักศึกษา
3. ความสัมพันธ์ (Relationship) หมายถึงความสัมพันธ์ระหว่างเอนทิตี โดยที่เอนทิตีหนึ่งสามารถมีความสัมพันธ์กับเอนทิตีอื่นเอนทิตีหนึ่งได้ เช่น นักศึกษาสามารถลงทะเบียนเรียนในหลายวิชา และวิชาใด ๆ ก็สามารถมีนักศึกษาหลายคนลงทะเบียนเรียนได้
4. คีย์หลัก (Primary Key) หมายถึงคีย์ที่ใช้เพื่อระบุเอนทิตีนั้น ๆ อย่างชัดเจน และไม่สามารถซ้ำกันได้
5. คีย์นอก (Foreign Key) หมายถึงคีย์ที่เป็นคีย์หลักของเอนทิตีหนึ่ง และเป็นคีย์ที่อยู่ในเอนทิตีอื่นเอนทิตีหนึ่ง

#### 2.2 ไมโครเซอร์วิส (Microservices)

Microservice[?] หรือ Microservice Architecture คือสถาปัตยกรรมการออกแบบ Service หรือก็คือออกแบบซอฟต์แวร์ โดยการที่ในชื่อมีคำว่า Micro นำหน้าอยู่ก็เพราะว่าเป็นการออกแบบที่ทำให้ Service มีขนาดเล็กเพื่อแก้ไขจุดด้อยของสถาปัตยกรรมการออกแบบอื่นๆ

##### 2.2.1 Monolithic VS Microservice

หาก Microservice เป็นการออกแบบ Service ให้มีขนาดเล็ก การจะเทียบให้เห็นภาพชัดเจนที่สุดก็ต้องเทียบกับ Monolithic ที่เป็นระบบที่มีขนาดใหญ่ โดย Monolithic จะเป็นระบบที่มีการทำงานทั้งหมดอยู่ใน Service เดียว



รูปที่ 2.1: รูปจาก nsights.daffodilsw.com

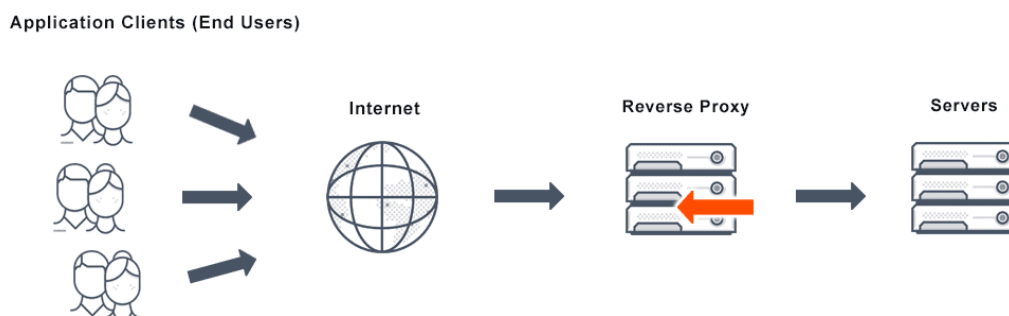
#### 2.2.1.1 ความแตกต่างระหว่าง Monolithic และ Microservice

- **Monolithic** เป็นชื่อของสถาปัตยกรรมการออกแบบซอฟต์แวร์หรือ Service ที่มีคนใช้งานเป็นจำนวนมากและมีมาอย่างยาวนาน โดยเป็นลักษณะของระบบที่การทำงานทุกอย่างจะรวมอยู่ในกลุ่มก้อนเดียวกัน และใช้งาน Database เดียวกัน (อย่างในภาพจะเห็นว่าเป็นเว็บไซต์ขายสินค้าที่มีฟังก์ชันจัดการผู้ใช้, ตะกร้าสินค้า และการส่งสินค้า รวมอยู่ด้วยกัน และใช้ฐานข้อมูลเดียวกัน)
- **Microservice** จะออกแบบโดยแยกการทำงานที่รวมกันเป็นก้อนใหญ่ๆของแบบ Monolithic ออกมาให้เล็กลงโดยอาจจะแยกตามบริการหรือตามฟังก์ชันการทำงานเลยก็ได้ (จากในภาพฟังก์ชันทั้งสามอย่างจะแยกออกจากกัน และไม่ได้ใช้ฐานข้อมูลเดียวกันในการเก็บข้อมูลอีกต่อไป เพราะแต่ละฟังก์ชันหรือบริการที่แยกออกมามีฐานข้อมูลเป็นของตัวเอง และสามารถติดต่อกันได้ผ่าน API )

#### 2.2.1.2 ข้อดีและข้อเสียของ Microservice

- ข้อดี
  1. การทำงานหลักแต่ละส่วนของระบบ ถ้าเป็นไปได้ควรแยกออกเป็น service แต่ละอัน เช่น จัดการสินค้า กับจัดการการซื้อสินค้าก็แยกกันไปเลย
  2. มีที่เก็บข้อมูลของตัวเอง
- ข้อเสีย
  1. การจัดการระบบที่มีหลาย service อาจจะทำให้การจัดการระบบทำได้ยากขึ้น
  2. การทำงานของระบบที่แยกออกมาอาจจะทำให้การทำงานของระบบช้าลง

## 2.3 Reverse Proxy



รูปที่ 2.2: รูปจาก <https://www.vmware.com/topics/reverse-proxy-server>

Reverse Proxy[?] เป็นเซิร์ฟเวอร์พร็อกซีที่ทำหน้าที่เป็นตัวกลางระหว่าง ไคลเอนต์ (Client) และ เซิร์ฟเวอร์ต้นทาง (Origin Server) โดยไคลเอนต์จะส่งคำขอ (Request) ไปยัง Reverse Proxy และจากนั้น Reverse Proxy จะส่งคำขอนั้นไปยังเซิร์ฟเวอร์ที่เหมาะสม แล้วรับคำตอบกลับมาเพื่อส่งต่อไปให้ไคลเอนต์

### 2.3.1 ประโยชน์ของการทำ Reverse Proxy

การทำ Reverse Proxy มีประโยชน์หลายอย่างโดยประโยชน์หลักๆ จะมีดังนี้

- ป้องกันการโจมตี DDoS
- ซ่อน IP จริงของเซิร์ฟเวอร์
- ใช้ SSL/TLS เพื่อเข้ารหัสข้อมูล
- กระจายโหลด (Load Balancing) ไปยังเซิร์ฟเวอร์หลายตัว
- แคชข้อมูล (Caching) ลดภาระของเซิร์ฟเวอร์ต้นทาง

## 2.4 Hypertext Transfer Protocol (HTTP)

HTTP (Hypertext Transfer Protocol) เป็นโปรโตคอลสื่อสารที่ใช้ในการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์บนเครือข่ายอินเทอร์เน็ต โดย HTTP มีหน้าที่เป็นตัวกลาง และเบราว์เซอร์ (web browsers) หรือแอปพลิเคชันอื่น ๆ ที่ใช้เครือข่ายอินเทอร์เน็ต ในการร้องขอและส่งข้อมูลระหว่างเว็บไซต์ (web servers)

## 2.5 Application Programming Interface (API)

คือ การเชื่อมต่อโปรแกรมประยุกต์ ในบริบทนี้ คำว่า "Application" หมายถึงทุกซอฟต์แวร์ที่มีฟังก์ชันชัดเจน และ "Interface" ก็คือตัวประสานหรือเป็นเหมือนสัญญาที่กำหนดวิธีการสื่อสารกันระหว่าง "Application" API ทำงานใน 4 รูปแบบด้วยกัน โดยขึ้นอยู่กับเวลาและสาเหตุที่สร้าง API

1. SOAP API - Simple Object Access Protocol (โปรโตคอลการเข้าถึงอ็อบเจกต์อย่างง่าย) ไคลเอนต์และเซิร์ฟเวอร์จะแลกเปลี่ยนข้อความโดยใช้ XML ซึ่งเป็น API ที่มีความยืดหยุ่นน้อยซึ่งเคยได้รับความนิยมมากกว่าในอดีต
2. RPC API - Remote Procedure Call (การเรียกใช้กระบวนการระยะไกล) ไคลเอนต์ดำเนินการฟังก์ชัน (หรือกระบวนการ) หนึ่งๆ บนเซิร์ฟเวอร์ และเซิร์ฟเวอร์ส่งผลลัพธ์กลับไปยังไคลเอนต์
3. Websocket API - Web API สมัยใหม่ที่ใช้อ็อบเจกต์ JSON ในการส่งข้อมูล WebSocket API รองรับการสื่อสารสองทางระหว่างแอปไคลเอนต์และเซิร์ฟเวอร์ เซิร์ฟเวอร์สามารถส่งข้อความเรียกกลับไปยังไคลเอนต์ที่เชื่อมต่อ จึงทำให้มีประสิทธิภาพมากกว่า REST API
4. REST API - API ที่ได้รับความนิยมและยืดหยุ่นที่สุดที่พบในเว็บไซด์ปัจจุบัน ไคลเอนต์ส่งคำขอไปยังเซิร์ฟเวอร์เป็นข้อมูล เซิร์ฟเวอร์ใช้ข้อมูลอินพุตจากไคลเอนต์นี้เพื่อเริ่มต้นฟังก์ชันภายในและส่งคืนข้อมูลเอาต์พุตกลับไปยังไคลเอนต์

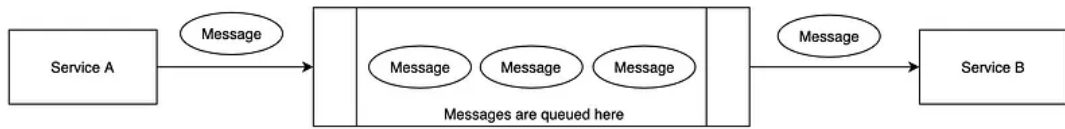
## 2.6 Json Web Token (JWT)

JWT[?]เป็นมาตรฐานแบบเปิด (RFC 7519) ที่กำหนดรูปแบบข้อมูลที่มีขนาดเล็กและสามารถตรวจสอบได้ในตัวเอง เพื่อใช้ในการส่งข้อมูลระหว่างฝ่ายต่างๆ อย่างปลอดภัยในรูปแบบของ JSON ข้อมูลนี้สามารถตรวจสอบและเชื่อถือได้ เนื่องจากการลงนามดิจิทัล (digitally signed)

## 2.7 Message Queue

Message Queue[?] (เรียกย่อๆว่า MQ) เป็นส่วนประกอบหนึ่งที่สำคัญในการออกแบบระบบขนาดใหญ่ โดย MQ ทำหน้าที่ในการรับ Message จากต้นทาง เก็บรักษาไว้ตามลำดับที่รับ Message เข้ามา และเปิดให้ปลายทาง มาหยิบ Message ออกไปที่ละ 1 Message (หรือมากกว่า) ตามลำดับที่กำหนดไว้ตามประเภทของ Queue นั่นๆโดยที่ MQ นั้นเอง ก็มีหลากหลายประเภท หลายยี่ห้อผู้ผลิต และหลากหลายลักษณะการใช้งาน แต่ในพื้นฐานแล้ว ก็จะมีลักษณะเหมือนกัน ลองมาดูภาพการทำงานแบบคร่าวๆกันครับ





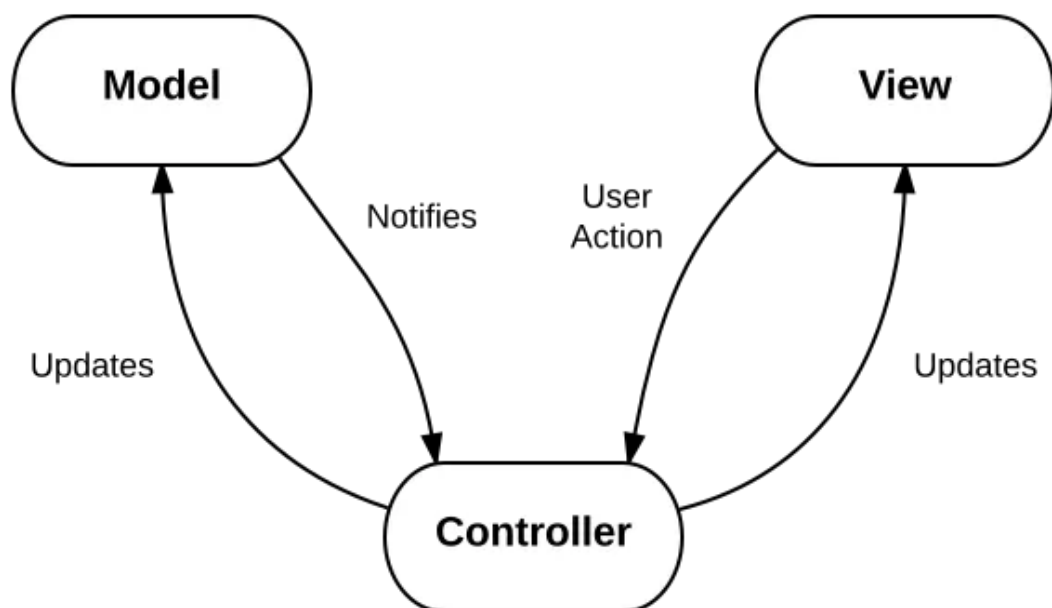
รูปที่ 2.3:

รูปจาก <https://panitw.medium.com/การใช้งาน-message-queue-pattern-65b90a6c4364>

จากภาพ เราเรียก Service A ว่า Producer (ผู้ผลิต) และ Service B ว่า Consumer (ผู้บริโภค) โดย Producer จะสร้าง Message และส่งเข้าไปรอไว้ในคิว เพื่อให้ Consumer มาหยิบข้อความไปใช้ ข้อความที่ส่งเข้าไปใน MQ ก็จะถูกเก็บรักษาเอาไว้รอให้ Consumer มาหยิบโดยความเร็วข้อความที่ส่งเข้ามา อาจจะไม่เท่ากับความเร็วจนข้อความที่ถูกดึงออกไป เช่นถ้า Producer ส่งข้อความทีละ 10 ข้อความต่อวินาที แต่ Consumer อ่านไปทำได้ทีละ 1 ข้อความต่อวินาที ก็จะทำให้มีข้อความค้างใน MQ เพิ่มขึ้นเรื่อยๆ 9 ข้อความต่อวินาที

## 2.8 Model-View-Controller (MVC)

MVC หรือ Model-View-Controller[?] เป็นรูปแบบการออกแบบโปรแกรมที่มีการแบ่งส่วนการทำงานออกเป็น 3 ส่วน ได้แก่ Model, View และ Controller เป็น Software Design Pattern หรือแนวทางการออกแบบซอฟต์แวร์รูปแบบหนึ่งของการเขียนซอฟต์แวร์



รูปที่ 2.4: รูปจาก <https://commons.wikimedia.org/wiki/File:MVC-basic.svg>

- **Model** คือส่วนที่เก็บข้อมูล และจัดการกับข้อมูล โดย Model จะไม่รู้จักร View จะส่งข้อมูลกลับไปยัง Controller เท่านั้น
- **View** คือส่วนที่แสดงผลข้อมูล และรับข้อมูลจากผู้ใช้ โดย View จะไม่รู้จักร Model แต่จะส่งข้อมูลกลับไปยัง Controller เท่านั้น
- **Controller** คือส่วนที่ควบคุมการทำงานของระบบ โดย Controller จะรับข้อมูลจาก View และส่งข้อมูลไปยัง Model และจาก Model ก็จะส่งข้อมูลกลับไปยัง View

### จุดประสงค์ของ MVC

- เพื่อแยกโค้ดออกเป็นส่วน ทำให้เราเปลี่ยนแปลงบางส่วนของโค้ดได้ โดยไม่กระทบกับส่วนอื่น ทำให้ Maintenance โค้ดที่หลังได้ง่าย
- ทำให้โค้ดสามารถถูกเขียนพร้อมกันโดยโปรแกรมเมอร์หลายคนได้
- สามารถนำโค้ดมาใช้ซ้ำได้

## 2.9 Elasticsearch

เนื่องด้วยการ Search project ด้วย full text search บน SQL Database ไม่ตอบโจทย์หากข้อมูลมีจำนวนมากและต้องการค้นหาจาก Text ที่มีความยาวมาก Elasticsearch จึงเข้ามาช่วยแก้ปัญหา ในส่วนนี้ โดย Elastic Search คือ Search Engine ที่ได้รับความนิยมอย่างแพร่หลาย ถูกออกแบบมาให้สามารถจัดเก็บ ค้นหา และวิเคราะห์ข้อมูลขนาดใหญ่ได้อย่างรวดเร็วและมีประสิทธิภาพ โดยใช้โครงสร้างแบบ Distributed (กระจายศูนย์) และ RESTful API เพื่อให้สามารถเข้าถึงและจัดการข้อมูลได้ง่าย คุณสมบัติของ Elastic search มีดังนี้ Full text search รองรับการค้นหาข้อมูลแบบเต็มรูปแบบและแม่นยำ, Distributed Architecture ทำงานแบบกระจายข้อมูลที่เรียกว่า sharding เพื่อรองรับโหลดขนาดใหญ่และเพิ่มประสิทธิภาพ, Scalability สามารถขยายขนาดได้ง่ายทั้งในรูปแบบ vertical scaling และ horizontal scaling , Near Real-Time Search สามารถค้นหาและดึงข้อมูลที่เพิ่งถูกเพิ่มเข้ามาได้อย่างรวดเร็ว, Integration รองรับการเชื่อมต่อกับ Logstash, Kibana และ Beats เพื่อสร้าง solution สำหรับการวิเคราะห์ข้อมูลและการ Monitor

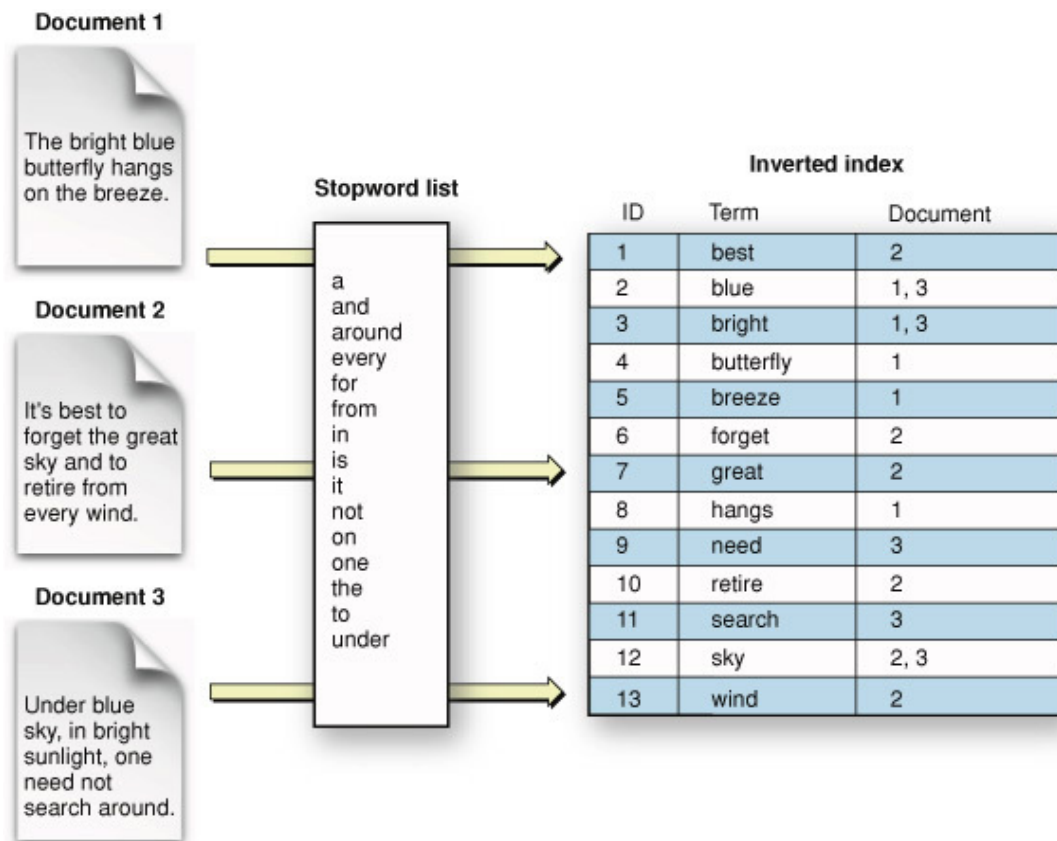
## 2.10 Inverted Index

Inverted Index เป็นโครงสร้างข้อมูลที่ถูกใช้ใน Elasticsearch เพื่อเพิ่มประสิทธิภาพในการค้นหาข้อมูลแบบ Full-Text Search โดยหลักการของ Inverted Index คือการสร้างดัชนีที่ทำให้สามารถค้นหาคำในเอกสารขนาดใหญ่ได้อย่างรวดเร็ว โครงสร้างของ Inverted Index มีดังนี้:

- **extbfTokenization** ข้อความจะถูกแยกออกเป็นคำย่อยๆ (Token) เช่น "Elasticsearch is fast" จะแยกออกเป็น ["elasticsearch", "is", "fast"].
- **extbfNormalization** คำจะถูกปรับรูปแบบ เช่น เปลี่ยนเป็นตัวพิมพ์เล็กทั้งหมด หรือกำจัดเครื่องหมายพิเศษ.

- extbfPosting List สร้างดัชนีโดยเก็บตำแหน่งของคำที่ปรากฏในเอกสาร เช่น "fast" อาจอยู่ในเอกสารที่ 1, 2 และ 5.

การใช้ Inverted Index ทำให้ Elasticsearch สามารถค้นหาข้อมูลได้เร็วขึ้นมากเมื่อเทียบกับการสแกนข้อมูลทั้งหมดแบบดั้งเดิมใน SQL Database.



รูปที่ 2.5: ตัวอย่างโครงสร้างของ Inverted Index

## 2.11 Kibana

Kibana เป็นเครื่องมือสำหรับแสดงผลและวิเคราะห์ข้อมูลที่ถูกจัดเก็บใน Elasticsearch โดยมีอินเทอร์เฟซแบบกราฟิกที่ช่วยให้ผู้ใช้สามารถสร้างแดชบอร์ด (Dashboard), ค้นหา, วิเคราะห์ และแสดงข้อมูลในรูปแบบต่างๆ ได้อย่างง่ายดาย Kibana มักถูกใช้ร่วมกับ Elasticsearch ในงานที่ต้องจัดเก็บ ค้นหา วิเคราะห์ และแสดงผลข้อมูล เช่น:

- การตรวจสอบเซิร์ฟเวอร์หรือแอปพลิเคชัน (Application Monitoring)
- ระบบค้นหาในเว็บไซต์หรือเอกสาร (Search Systems)
- การสร้างแดชบอร์ดสรุปข้อมูลธุรกิจ (Business Analytics)
- การตรวจจับภัยคุกคาม (Security Monitoring)

## 2.12 Spring Boot

Spring Boot เป็นเฟรมเวิร์กที่พัฒนาต่อยอดจาก Spring Framework โดยมีจุดเด่นคือทำให้การตั้งค่าซับซ้อนของ Spring ง่ายขึ้น และช่วยให้พัฒนาเว็บแอปพลิเคชัน หรือ REST API ได้เร็วขึ้น คุณสมบัติของ Spring Boot มีดังนี้:

- **Auto Configuration** – ตั้งค่าเริ่มต้นอัตโนมัติ เช่น Database, Security, Logging
- **Embedded Server** – มี Web Server ในตัว (Tomcat, Jetty, Undertow)
- **Spring Boot Starter** – ใช้ dependency สำเร็จรูป ทำให้ง่ายต่อการใช้งานเครื่องมือต่างๆ เช่น พัฒนาเว็บแอปและ REST API
- **Spring Boot Actuator** – ใช้ตรวจสอบสถานะระบบ เช่น health check
- **Microservices Ready** – รองรับการพัฒนา Microservices ได้ง่าย

## 2.13 Hibernate

Hibernate เป็นเฟรมเวิร์กที่ใช้สำหรับ ORM (Object-Relational Mapping) ใน Java ซึ่งช่วยให้สามารถทำงานกับฐานข้อมูลได้สะดวกขึ้นโดยไม่ต้องเขียน SQL โดยตรง คุณสมบัติหลักของ Hibernate มีดังนี้:

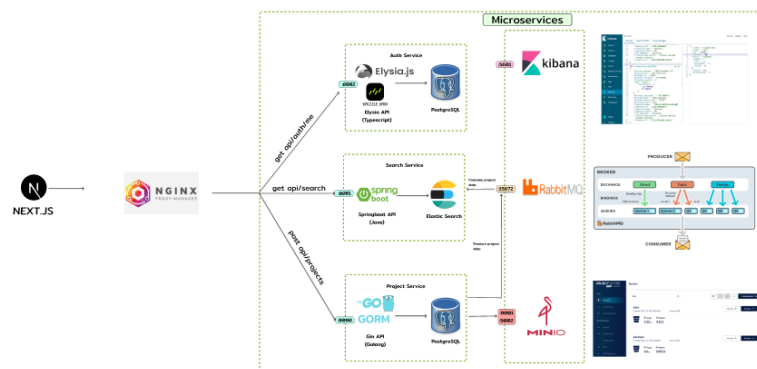
- **Mapping Object กับ Database** – แปลง Java Object ให้สามารถบันทึกลงฐานข้อมูลได้อัตโนมัติ
- **HQL (Hibernate Query Language)** – ภาษา Query ที่สามารถใช้แทน SQL และเป็นอิสระจากฐานข้อมูล
- **Lazy Loading** – โหลดข้อมูลเฉพาะส่วนที่จำเป็นเพื่อลดการใช้ทรัพยากร
- **Transaction Management** – รองรับการทำงานกับ Transaction และ ACID properties
- **Database Independence** – ใช้งานได้กับหลายฐานข้อมูล เช่น MySQL, PostgreSQL, Oracle

## บทที่ 3

### โครงสร้างและขั้นตอนการทำงาน

#### 3.1 สถาปัตยกรรมระบบ

โครงการนี้ได้ออกแบบสถาปัตยกรรมของระบบเป็นแบบ **Microservices** โดยบทนี้จะกล่าวถึงโครงสร้างของระบบทั้งหมด และอธิบายถึงแต่ละส่วนของระบบ โดยระบบจะถูกแบ่งออกเป็นส่วนย่อยๆ แต่ละส่วนจะมีหน้าที่และความรับผิดชอบในการทำงานที่แตกต่างกัน



รูปที่ 3.1: รูปภาพแสดงสถาปัตยกรรมของระบบ

จากรูปที่ 3.1 แสดงถึงสถาปัตยกรรมของระบบที่ถูกแบ่งออกเป็นส่วนย่อยๆ ดังนี้

1. **Frontend** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการแสดงผลของระบบ โดยส่วนนี้จะถูกพัฒนาด้วย Next.js
2. **Reverse Proxy** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการจัดการการเชื่อมต่อระหว่าง Frontend และ Backend โดยส่วนนี้จะถูกพัฒนาด้วย Nginx Proxy Manager(NPM)
3. **Backend** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการจัดการข้อมูลและการทำงานของระบบ โดยส่วนนี้จะถูกแบ่งออกเป็นส่วนย่อยๆ ดังนี้
  - **Auth Service** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการจัดการข้อมูลของผู้ใช้งาน โดยส่วนนี้จะถูกพัฒนาด้วย Typescript Elysia.js
  - **Project Service** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการจัดการข้อมูลของโครงการ โดยส่วนนี้จะถูกพัฒนาด้วย Go Gin
  - **Search Service** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการค้นหาข้อมูลของโครงการ โดยส่วนนี้จะถูกพัฒนาด้วย Java Sprint Boot & Elasticsearch

#### 3.2 Frontend (Waiting for edit)

ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการแสดงผลของระบบ โดยส่วนนี้จะถูกพัฒนาด้วย Next.js โดยส่วนนี้จะถูกแบ่งออกเป็นส่วนย่อยๆ ดังนี้

- **Home Page** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการแสดงผลหน้าแรกของเว็บไซต์
- **Project Page** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการแสดงผลหน้าโครงการ
- **Search Page** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการแสดงผลหน้าค้นหา
- **Profile Page** ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการแสดงผลหน้าโปรไฟล์

### 3.3 Reverse Proxy

ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการจัดการการเชื่อมต่อระหว่าง **Frontend** และ **Backend** โดยส่วนนี้จะถูกพัฒนาด้วย **Nginx Proxy Manager(NPM)[?]** มีหน้าที่คือ การแปลง ip address ของ server ให้เป็น domain name ยกตัวอย่างเช่น แปลง IP Address 139.59.117.147:8080 → domainname.com โดยส่วนนี้จะทำให้เราสามารถเข้าถึงเว็บไซต์ผ่าน domainname.com ได้ และยังสามารถจัดการการเชื่อมต่อระหว่าง **Frontend** และ **Backend** ได้อีกด้วย

**New Proxy Host** [X]

⚡ Details Custom locations SSL ⚙️ Advanced

**Domain Names \***

itc.d...go.th

Scheme *	Forward Hostname / IP *	Forward Port *
http	192.168.1.250	8999

☐ Cache Assets
 ☒ Block Common Exploits
 ☐ Websockets Support

**Access List**

Publicly Accessible

Cancel Save

รูปที่ 3.2: รูปภาพแสดงการทแปลง IP ของ Nginx Reverse Proxy

อีก feature หนึ่งที่สำคัญของ Nginx Reverse Proxy คือการจัดการ SSL Certificate โดย Reverse Proxy จะทำหน้าที่ในการจัดการ SSL Certificate ให้กับเว็บไซต์ ซึ่งจะทำให้เว็บไซต์มีความ

New Proxy Host

×

⚡ Details

📍 Custom locations

🔒 SSL

⚙️ Advanced

SSL Certificate

Request a new SSL Certificate

❌ None

This host will not use HTTPS

✅ Request a new SSL Certificate

with Let's Encrypt

🛡️ itc. ██████████.th

Let's Encrypt – Expires: 26th September 2022, 8:31 am

🛡️ ██████████

██████████

🛡️ ██████████

██████████

☐ I Agree to the [Let's Encrypt Terms of Service](#) \*

Cancel

Save

รูปที่ 3.3: รูปภาพแสดงการจัดการ SSL Certificate ของ Nginx Reverse Proxy

ปลอดภัยมากขึ้น และยังสามารถใช้งานได้ทั้งบน http และ https อีกด้วย

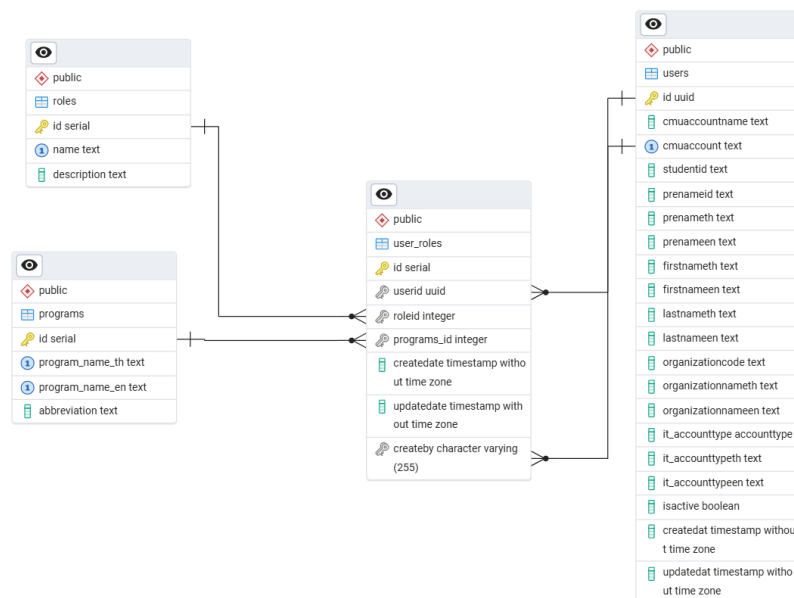
## 3.4 Backend

ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการจัดการข้อมูลและการทำงานของระบบ โดยส่วนนี้จะถูกแบ่งออกเป็นส่วนย่อยๆ ดังนี้

### 3.4.1 Auth Service

ส่วนนี้เป็นส่วนที่ทำหน้าที่ในการจัดการข้อมูลของผู้ใช้งาน โดยส่วนนี้จะถูกพัฒนาด้วย Typescript Elysia.js โดยส่วนนี้จะมีหน้าที่คือ จัดการข้อมูลของผู้ใช้งานทำหน้าที่ในการเข้าสู่ระบบ และจัดการสิทธิ์ของผู้ใช้งาน(Role) โดยมีการใช้งาน OAuth2 ในการจัดการการเข้าสู่ระบบ และมีการใช้งาน JWT เพื่อสร้าง Token ส่งกลับไปยัง Frontend

### 3.4.1.1 Database Schema



รูปที่ 3.4: รูปภาพแสดง Database Schema ของ Auth Service

#### โครงสร้างของฐานข้อมูล

- **users** ตารางนี้จะเป็นตารางที่เก็บข้อมูลของผู้ใช้งาน
- **roles** ตารางนี้จะเป็นตารางที่เก็บข้อมูลของตำแหน่งของผู้ใช้งาน
- **programs** ตารางนี้จะเป็นตารางที่เก็บข้อมูลของโปรแกรมหรือหลักสูตรที่มีอยู่ในระบบ
- **user\_roles** ตารางนี้จะแสดงถึงความสัมพันธ์ของผู้ใช้งานกับตำแหน่งของผู้ใช้งานในโปรแกรมหรือหลักสูตรนั้นๆ

### 3.4.2 Project Service

### 3.4.3 Search Service



## บทที่ 4

### การทดลองและผลลัพธ์

ในบทนี้จะทำการทดสอบการทำงานของระบบในฟังก์ชันหลักต่างๆ

#### 4.1 การทดสอบฟังก์ชันการทำงาน

##### 4.1.1 การเข้าสู่ระบบของผู้ใช้

- **วัตถุประสงค์:** เพื่อตรวจสอบว่าผู้ใช้งานสามารถเข้าสู่ระบบได้โดยใช้บัญชี CMU Account
- **ขั้นตอนการทดสอบ:**
  1. ไปยังหน้าเข้าสู่ระบบ
  2. ป้อนข้อมูลบัญชี CMU Account ที่ถูกต้อง
  3. คลิกปุ่ม "เข้าสู่ระบบ"
- **ผลลัพธ์ที่คาดหวัง:** ผู้ใช้งานสามารถเข้าสู่ระบบได้สำเร็จ และถูกเปลี่ยนเส้นทางไปยังหน้าแดชบอร์ด
- **ผลลัพธ์ที่ได้:** ผู้ใช้งานสามารถเข้าสู่ระบบได้สำเร็จ และถูกเปลี่ยนเส้นทางไปยังหน้าแดชบอร์ด
- **สถานะ:** ผ่าน

##### 4.1.2 การค้นหาโครงการงาน

##### 4.1.2.1 การค้นหาแบบ Quick Search

- **วัตถุประสงค์:** เพื่อตรวจสอบว่าผู้ใช้งานสามารถค้นหาโครงการงานได้โดยใช้ฟังก์ชันการค้นหาแบบ Quick Search
- **ขั้นตอนการทดสอบ:**
  1. ไปยังหน้าค้นหา
  2. ป้อนคำค้นหาในช่องค้นหา (เช่น ชื่อโครงการงาน, คำอธิบาย, รหัสนักศึกษา/ชื่อ, ชื่ออาจารย์ที่ปรึกษา ภาษาไทย/อังกฤษ)
  3. คลิกปุ่ม "ค้นหา"
- **ผลลัพธ์ที่คาดหวัง:** โครงการงานที่เกี่ยวข้องกับคำค้นหาจะแสดงขึ้นมา
- **ผลลัพธ์ที่ได้:** โครงการงานที่เกี่ยวข้องแสดงผลได้อย่างถูกต้อง แต่มีปัญหาคือไม่สามารถเว้นคำในการค้นหา เช่น ค้นหาคำว่า "Digital Platform" แต่ถ้าพิมพ์ "Digital" หรือ "Platform" จะหาเจอ
- **สถานะ:** ผ่านบางส่วน

#### 4.1.2.2 การค้นหาแบบ Advanced Search

- **วัตถุประสงค์:** เพื่อตรวจสอบว่าผู้ใช้งานสามารถค้นหาโครงการได้โดยใช้ฟังก์ชันการค้นหาแบบ Advanced Search
- **ขั้นตอนการทดสอบ:**
  1. ไปยังหน้าค้นหา
  2. เลือกหมวดหมู่ต่างๆ สำหรับการค้นหา (เช่น หมวดหมู่โครงการ, ปีการศึกษา, ชื่ออาจารย์ที่ปรึกษา)
  3. คลิกปุ่ม "ค้นหา"
- **ผลลัพธ์ที่คาดหวัง:** โครงการที่เกี่ยวข้องกับหมวดหมู่ที่เลือกจะแสดงขึ้นมา
- **ผลลัพธ์ที่ได้:** โครงการที่เกี่ยวข้องแสดงผลได้อย่างถูกต้อง
- **สถานะ:** ผ่าน

#### 4.1.2.3 การค้นหาแบบ PDF Search

- **วัตถุประสงค์:** เพื่อตรวจสอบว่าผู้ใช้งานสามารถค้นหาโครงการได้โดยใช้ฟังก์ชันการค้นหาแบบ PDF Search
- **ขั้นตอนการทดสอบ:**
  1. ไปยังหน้าค้นหา
  2. ป้อนคำค้นหาในช่องค้นหา
  3. คลิกปุ่ม "ค้นหา"
- **ผลลัพธ์ที่คาดหวัง:** โครงการที่มีคำค้นหาในรายงาน PDF จะแสดงขึ้นมาและมีการไฮไลต์คำค้นหา
- **ผลลัพธ์ที่ได้:** โครงการที่เกี่ยวข้องแสดงผลได้อย่างถูกต้อง แต่การค้นหาคำภาษาไทยยังมีผิดพลาดบ้าง เช่น ค้นหาว่า "บริหาร" อาจจะมีคำว่า "บริการ" มาด้วย ซึ่งคาดว่าปัญหาน่าจะมาจาก Elasticsearch
- **สถานะ:** ผ่านบางส่วน

#### 4.1.2.4 การค้นหาแบบ Keyword Search

- **วัตถุประสงค์:** เพื่อตรวจสอบว่าผู้ใช้งานสามารถค้นหาโครงการได้โดยใช้ฟังก์ชันการค้นหาแบบ Keyword Search
- **ขั้นตอนการทดสอบ:**
  1. ไปยังหน้าค้นหา
  2. ป้อนคำค้นหาในช่องค้นหา

### 3. คลิกปุ่ม ”ค้นหา”

- ผลลัพธ์ที่คาดหวัง: โครงการที่เกี่ยวข้องกับคำค้นหาคะแสดงขึ้นมา
- ผลลัพธ์ที่ได้: โครงการที่เกี่ยวข้องแสดงผลได้อย่างถูกต้อง
- สถานะ: ผ่าน

#### 4.1.3 การสร้างโครงการ

- วัตถุประสงค์: เพื่อตรวจสอบว่าผู้ใช้งานสามารถสร้างโครงการใหม่ได้
- ขั้นตอนการทดสอบ:
  1. ไปยังหน้าแดชบอร์ด
  2. คลิกปุ่ม ”Create Project”
  3. กรอกรายละเอียดโครงการ
  4. คลิกปุ่ม ”Save”
- ผลลัพธ์ที่คาดหวัง: โครงการถูกสร้างสำเร็จ และแสดงในหน้าแดชบอร์ด
- ผลลัพธ์ที่ได้: โครงการถูกสร้างสำเร็จและแสดงผล
- สถานะ: ผ่าน

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### 5.1 สรุปผล

ในบทนี้จะสรุปถึงข้อจำกัดของระบบในด้านต่างๆ ที่ระบบมีในเนื้อหาส่วนนี้ด้วย

- ระบบสามารถรวบรวมและจัดเก็บข้อมูลโปรเจกต์จบของนักศึกษาได้อย่างมีประสิทธิภาพ
- ผู้ใช้สามารถค้นหาและเข้าถึงข้อมูลโปรเจกต์ได้ง่ายขึ้น
- ระบบมีการจัดการสิทธิ์การเข้าถึงข้อมูลที่ดี ทำให้ข้อมูลมีความปลอดภัย
- อย่างไรก็ตาม ระบบยังมีข้อจำกัดในการแสดงผลบนอุปกรณ์มือถือ และการจัดการข้อมูลที่มีขนาดใหญ่

#### 5.2 ปัญหาที่พบและแนวทางการแก้ไข

ในการทำโครงงานนี้ พบว่าเกิดปัญหาหลักๆ ดังนี้

- การจัดการข้อมูลที่มีขนาดใหญ่และการค้นหาข้อมูลที่มีประสิทธิภาพยังคงเป็นปัญหา
- การแสดงผลบนอุปกรณ์มือถือยังไม่รองรับอย่างเต็มที่
- การค้นหาข้อมูลจากในโครงงานยังไม่สามารถแสดงผลได้ว่าอยู่ส่วนไหนของโครงงาน

#### 5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

ข้อเสนอแนะเพื่อพัฒนาโครงงานนี้ต่อไป มีดังนี้

- ปรับปรุงการแสดงผลให้รองรับการใช้งานบนอุปกรณ์มือถือ
- พัฒนาระบบการจัดการข้อมูลให้มีประสิทธิภาพมากขึ้น โดยใช้เทคโนโลยีการจัดการข้อมูลที่เหมาะสม
- ปรับปรุงการจัดการสิทธิ์การเข้าถึงข้อมูลให้ใช้งานง่ายและมีความปลอดภัยมากขึ้น
- เพิ่มฟีเจอร์การค้นหาข้อมูลที่มีประสิทธิภาพและรวดเร็ว เช่นการเพิ่มการค้นหาจาก keyword

ภาคผนวก

## ภาคผนวก ก

### คู่มือการติดตั้ง

#### ก.1 Next.js 14 (Frontend)



โครงสร้างของโปรเจกต์นี้ประกอบด้วยหลายไฟล์และโฟลเดอร์ที่สำคัญสำหรับการพัฒนาและการตั้งค่าโปรเจกต์ โดยมีรายละเอียดดังนี้:

##### ก.1.1 Main Files and Folders

- **.dockerignore:** ไฟล์ที่ระบุไฟล์หรือโฟลเดอร์ที่ไม่ต้องการให้ Docker คัดลอกไปยัง image
- **.env, .env.local, .env.prod:** ไฟล์ที่ใช้เก็บค่าตัวแปรสภาพแวดล้อม (environment variables) สำหรับการตั้งค่าโปรเจกต์ในสภาพแวดล้อมต่างๆ
- **.eslinttrc.json:** ไฟล์การตั้งค่า ESLint สำหรับการตรวจสอบและจัดรูปแบบโค้ด
- **.gitignore:** ไฟล์ที่ระบุไฟล์หรือโฟลเดอร์ที่ไม่ต้องการให้ Git ติดตาม
- **docker-compose.yml:** ไฟล์การตั้งค่า Docker Compose สำหรับการจัดการ container หลายๆ ตัว
- **Dockerfile:** ไฟล์การตั้งค่า Docker สำหรับการสร้าง Docker image
- **next-env.d.ts:** ไฟล์การตั้งค่า TypeScript สำหรับโปรเจกต์ Next.js
- **next.config.mjs:** ไฟล์การตั้งค่า Next.js
- **package.json:** ไฟล์ที่เก็บข้อมูลเกี่ยวกับโปรเจกต์ Node.js รวมถึง dependencies และสคริปต์ต่างๆ
- **postcss.config.mjs:** ไฟล์การตั้งค่า PostCSS
- **README.md:** ไฟล์เอกสารสำหรับโปรเจกต์
- **tailwind.config.ts:** ไฟล์การตั้งค่า Tailwind CSS
- **tsconfig.json:** ไฟล์การตั้งค่า TypeScript

### ก.1.2 Subfolders

- **workflows:** โฟลเดอร์ที่เก็บไฟล์การตั้งค่า GitHub Actions สำหรับการทำงานอัตโนมัติ
- **.next:** โฟลเดอร์ที่เก็บไฟล์ที่ถูกสร้างขึ้นโดย Next.js หลังจากการ build
- **.vs:** โฟลเดอร์ที่เก็บการตั้งค่าของ Visual Studio Code
- **app:** โฟลเดอร์ที่เก็บไฟล์และโฟลเดอร์ที่เกี่ยวข้องกับการทำงานของแอปพลิเคชัน
- **components:** โฟลเดอร์ที่เก็บคอมโพเนนต์ React ที่ใช้ในโปรเจกต์
- **config:** โฟลเดอร์ที่เก็บไฟล์การตั้งค่าต่างๆ
- **context:** โฟลเดอร์ที่เก็บไฟล์ context ของ React
- **dtos:** โฟลเดอร์ที่เก็บไฟล์ Data Transfer Objects (DTOs)
- **hooks:** โฟลเดอร์ที่เก็บ custom hooks ของ React
- **models:** โฟลเดอร์ที่เก็บไฟล์โมเดลต่างๆ
- **pages:** โฟลเดอร์ที่เก็บไฟล์เพจของ Next.js
- **public:** โฟลเดอร์ที่เก็บไฟล์สาธารณะ เช่น รูปภาพ
- **types:** โฟลเดอร์ที่เก็บไฟล์ประเภท (types) ของ TypeScript
- **utils:** โฟลเดอร์ที่เก็บไฟล์ utility functions

## ก.2 Installation Guide

### ก.2.1 Clone the Project from GitHub

```
git clone https://github.com/259492-ProjectBox/frontend-projectbox.git  
cd <repository-directory>
```

### ก.2.2 Install Dependencies

```
npm install
```

### ก.2.3 Set Up Environment Variables

สร้างไฟล์ `.env.local` และกำหนดค่าตัวแปรสภาพแวดล้อมตามที่ต้องการ

### ก.2.4 Run the Project in Development Mode

```
npm run dev
```

### ก.2.5 Build Docker Image (Optional)

```
docker build -t <image-name> .
```

### ก.3 Elysia (Auth\_service)



#### ก.3.1 Project Structure

รายละเอียดของไฟล์และโฟลเดอร์หลัก:

- **.env:** ไฟล์นี้ใช้สำหรับเก็บค่าตัวแปรสภาพแวดล้อม เช่น ข้อมูลการเชื่อมต่อฐานข้อมูล และคีย์ลับต่างๆ
- **Dockerfile:** ไฟล์นี้ใช้สำหรับสร้าง Docker image ของโปรเจกต์
- **docker-compose.yml** และ **docker-compose.prod.yml:** ไฟล์นี้ใช้สำหรับการตั้งค่า Docker Compose เพื่อรันบริการต่างๆ ที่จำเป็นสำหรับโปรเจกต์
- **drizzle/:** โฟลเดอร์นี้เก็บไฟล์ที่เกี่ยวข้องกับการจัดการฐานข้อมูล เช่น สคิมาและการย้ายข้อมูล
- **src/:** โฟลเดอร์นี้เก็บโค้ดหลักของโปรเจกต์ แบ่งออกเป็นหลายโฟลเดอร์ย่อย เช่น **controllers**, **dtos**, **middleware**, **repositories**, **services**
- **types/:** โฟลเดอร์นี้เก็บไฟล์ที่ประกาศประเภทข้อมูลต่างๆ
- **utils/:** โฟลเดอร์นี้เก็บไฟล์ที่มีฟังก์ชันช่วยเหลือต่างๆ

#### ก.3.2 Installation and Usage

##### 1. ติดตั้ง Bun:

Bun Installation  
<https://bun.sh/docs/installation>

##### 2. ติดตั้ง Dependencies:

```
bun install
```

##### 3. ตั้งค่าฐานข้อมูล:

- ตรวจสอบให้แน่ใจว่าไฟล์ **.env** มีค่าตัวแปรสภาพแวดล้อมที่ถูกต้องสำหรับการเชื่อมต่อฐานข้อมูล



- รันคำสั่งต่อไปนี้เพื่อสร้างและตั้งค่าฐานข้อมูล:

#### 4. รันโปรเจกต์ในโหมดพัฒนา:

```
bun run dev
```

### ก.4 Golang Project (Project\_service)



#### ก.4.1 Project Structure

รายละเอียดของไฟล์และโฟลเดอร์หลัก:

- **.env.example:** ไฟล์นี้ใช้สำหรับเป็นตัวอย่างไฟล์สำหรับไฟล์ .env ในการเก็บค่าตัวแปรสภาพแวดล้อม เช่น ข้อมูลการเชื่อมต่อฐานข้อมูล และคีย์ลับต่างๆ
- **Dockerfile:** ไฟล์นี้ใช้สำหรับสร้าง Docker image ของโปรเจกต์
- **docker-compose.yml และ docker-compose.prod.yml:** ไฟล์นี้ใช้สำหรับการตั้งค่า Docker Compose เพื่อรันบริการต่างๆ ที่จำเป็นสำหรับโปรเจกต์
- **configs/:** โฟลเดอร์นี้เก็บ config ของ โปรเจกต์
- **db/:** โฟลเดอร์นี้เก็บไฟล์ที่เกี่ยวข้องกับการจัดการฐานข้อมูล เช่น สคิมาและการย้ายข้อมูล การเชื่อมต่อฐานข้อมูลต่างๆ
- **docs/:** โฟลเดอร์นี้เก็บไฟล์ API Documentation ของโปรเจกต์
- **routers/:** โฟลเดอร์นี้เก็บไฟล์ router ของโปรเจกต์เพื่อ routing request จาก client
- **handlers/:** โฟลเดอร์นี้เก็บไฟล์ handler ของโปรเจกต์สำหรับรับส่ง request จาก client
- **models/:** โฟลเดอร์นี้เก็บไฟล์โมเดลต่างๆ
- **services/:** โฟลเดอร์นี้เก็บไฟล์ที่เกี่ยวข้องกับการจัดการ logic ของโปรเจกต์
- **repositories/:** โฟลเดอร์นี้เก็บไฟล์ที่เกี่ยวข้องกับการจัดการข้อมูลในฐานข้อมูล
- **queues/:** โฟลเดอร์นี้เก็บไฟล์ที่เกี่ยวข้องกับการจัดการ queue ของโปรเจกต์
- **utils/:** โฟลเดอร์นี้เก็บไฟล์ที่มีฟังก์ชันช่วยเหลือต่างๆ

## ก.4.2 Installation and Usage

### 1. ติดตั้ง Golang:

Golang Installation  
<https://go.dev/doc/install>

### 2. ติดตั้ง Dependencies:

```
go mod tidy
```

### 3. ติดตั้ง Dependencies เพิ่มเติม:

- go install github.com/air-verse/air@latest
- go install github.com/swaggo/swag/cmd/swag@latest:
- scoop install poppler:

### 4. ตั้งค่าฐานข้อมูล:

- ตรวจสอบให้แน่ใจว่าไฟล์ .env มีค่าตัวแปรสภาพแวดล้อมที่ถูกต้องสำหรับการเชื่อมต่อฐานข้อมูล
- รันคำสั่งต่อไปนี้เพื่อสร้าง container ของฐานข้อมูล:

```
docker compose up -d
```

### 5. รันโปรเจกต์:

```
air
```

## ก.5 Java Project (Search\_service)



### ก.5.1 Project Structure

รายละเอียดของไฟล์และโฟลเดอร์หลัก:

- **configs/**: โฟลเดอร์นี้เก็บ config ของโปรเจกต์ เช่น Cors, RabbitMQ
- **consumers/**: โฟลเดอร์นี้เก็บไฟล์ที่เกี่ยวข้องกับการ consume message จาก Message Broker เช่น RabbitMQ
- **controllers/**: โฟลเดอร์นี้เก็บไฟล์ handler ของโปรเจกต์สำหรับรับส่ง request จาก client
- **db/**: โฟลเดอร์นี้เก็บไฟล์ที่เกี่ยวข้องกับการจัดการฐานข้อมูล เช่น การเชื่อมต่อฐานข้อมูลต่างๆ
- **models/**: โฟลเดอร์นี้เก็บไฟล์โมเดลต่างๆ
- **dtos/**: โฟลเดอร์นี้เก็บไฟล์ dto สำหรับการจัดการสร้าง object ของโปรเจกต์
- **services/**: โฟลเดอร์นี้เก็บไฟล์ที่เกี่ยวข้องกับการจัดการ logic ของโปรเจกต์
- **repositories/**: โฟลเดอร์นี้เก็บไฟล์ที่เกี่ยวข้องกับการจัดการข้อมูลในฐานข้อมูล

### ก.5.2 Installation and Usage

#### 1. ติดตั้ง Java:

Java Installation

[https://www.java.com/en/download/help/windows\\_manual\\_download.html](https://www.java.com/en/download/help/windows_manual_download.html)

#### 2. ติดตั้ง Dependencies:

```
mvn clean install
```

### 3. ตั้งค่าฐานข้อมูล:

- ตรวจสอบให้แน่ใจว่าไฟล์ `src/main/resources/application-dev.properties` มีค่าตัวแปรสภาพแวดล้อมที่ถูกต้องสำหรับการเชื่อมต่อฐานข้อมูล
- รันคำสั่งต่อไปนี้เพื่อสร้าง `container` ของฐานข้อมูล:

```
docker compose up -d
```

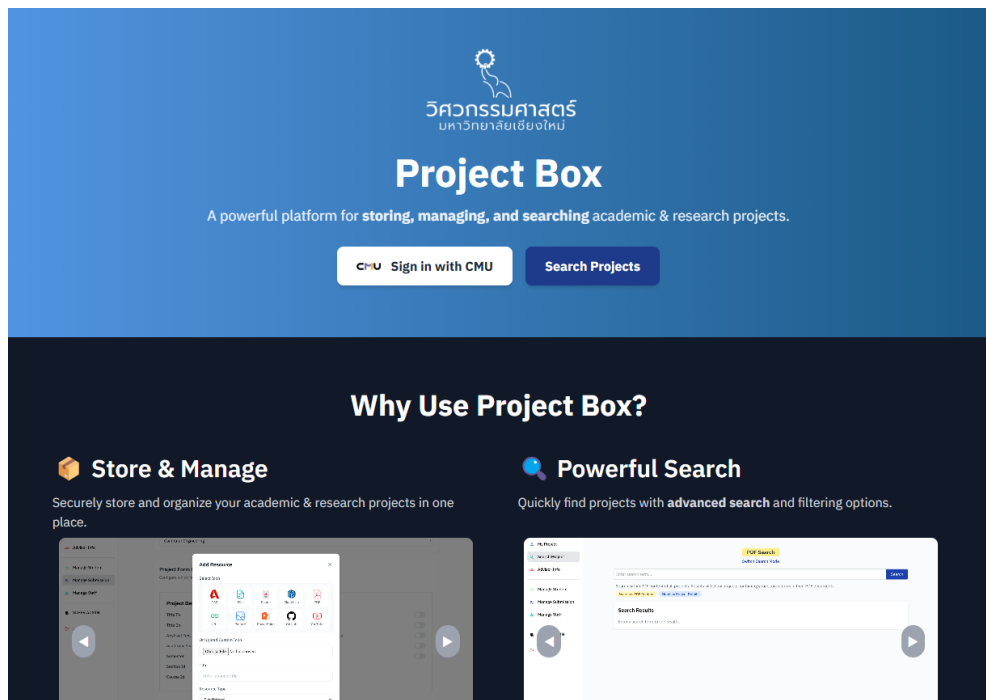
### 4. รันโปรเจกต์:

```
mvn spring-boot:run
```

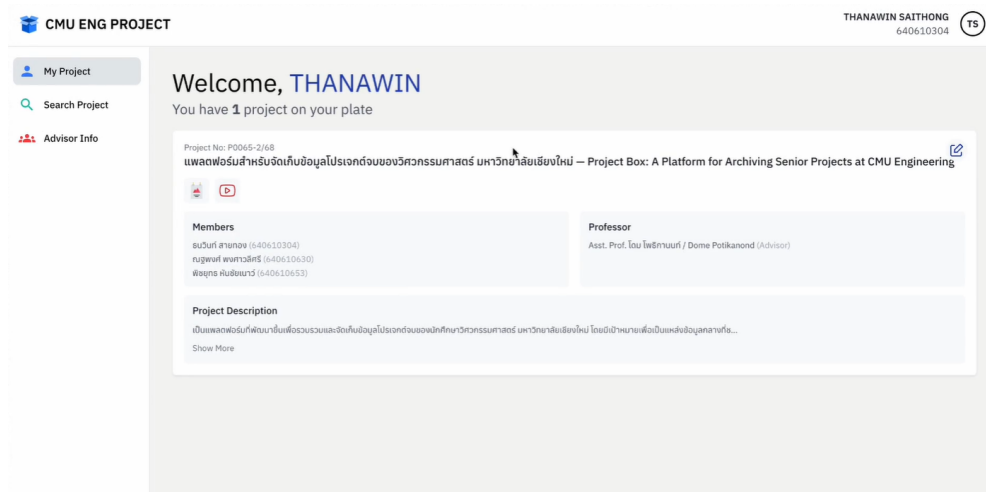
## ภาคผนวก ข คู่มือการใช้งานระบบ

### • การใช้งานในส่วนของ User ทั่วไป

1. ทำการ Login เข้าสู่ระบบผ่าน CMU Account หรือ สามารถค้นหาโปรเจกต์ได้โดยไม่ต้อง Login



2. เมื่อทำการ Login สำเร็จ จะเข้าสู่หน้า Dashboard ที่แสดงโปรเจกต์ที่เคยสร้างไว้แล้ว หรือ สามารถสร้างโปรเจกต์ใหม่ได้เมื่อมีการลงทะเบียนเข้าใช้งาน



### 3. การค้นหาโปรเจ็ค สามารถค้นหาโปรเจ็คที่ต้องการได้จากช่องค้นหาโครงการ

#### (a) Quick Search ใช้การค้นหาโครงการแบบง่าย

The screenshot shows the 'CMU ENG PROJECT' web application. On the left sidebar, there are three menu items: 'My Project', 'Search Project' (highlighted with a mouse cursor), and 'Advisor Info'. The main content area features a search bar with the placeholder text 'Enter search term...'. Above the search bar are three tabs: 'Quick Search' (active), 'Advanced Search', and 'PDF Search'. Below the search bar, there is a row of four colored badges: 'Project Title TH/EN' (blue), 'Description' (green), 'Student ID/Name' (orange), and 'Advisor Name TH/EN' (purple). A 'Search' button is located to the right of the search bar.

#### (b) Advanced Search ใช้การค้นหาโครงการแบบละเอียด

The screenshot shows the 'CMU ENG PROJECT' web application with the 'Advanced Search' tab selected. The search bar now contains a search term. Below the search bar, there are several input fields and dropdown menus for filtering results: 'Course No.' (with example 'e.g. 261492'), 'Student No.' (with example 'e.g. 640610633'), 'Academic Year' (with example 'e.g. 2568'), 'Program' (a dropdown menu with 'Select Program'), 'Project Title (EN or TH)' (with example 'e.g. Project Box'), 'Advisor Name (EN or TH)' (with example 'e.g. Pichayoot Hunchainao'), and 'Semester' (a dropdown menu with 'Select Semester'). A 'Search' button is located at the bottom right of the form.

#### (c) PDF Search ใช้การค้นหาโครงการจากไฟล์ PDF

The screenshot shows the 'CMU ENG PROJECT' web application with the 'PDF Search' tab selected. The search bar contains the text 'สอู'. Below the search bar, there is a message: 'Search within PDF content of all projects. Results will show projects containing your search terms in their PDF documents.' Below this message, there are two tabs: 'Searches PDF Content' (active) and 'Matches Project Details'. The main content area displays the search results for the project 'สอู'. The results include the project title 'สอู: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง – Sreetner: street scanner system for urban asset management', the project number 'Project No. P0043-2/68', the members list (อาจารย์สุวิมล (640610626), อธิวิมล (640610634), สุวิมล (640610639)), the professor list (Assoc. Prof. Dr. สอู พิทักษ์บุญ / Sani Phithakitkun (Advisor), Asst. Prof. Dr. อนุช อนุช / Ann Patanajom (Committee), Asst. Prof. Dr. อนุช อนุช / Navadon Khuntitgit (Committee)), and the project description 'โครงการ Sreetner (Street Scanner System for Urban Asset Management) เป็นโครงการที่พัฒนาขึ้นเพื่ออำนวยความสะดวกในการจัดการสินทรัพย์เมือง...'. A 'Show More' link is located at the bottom of the project description.

## (d) Keyword Search ใช้การค้นหาโครงการจาก Keyword

Quick Search Advanced Search PDF Search Keyword Search

AI Search

Search projects by keyword. Results will show projects containing your search terms in their keywords.

### Search Results

Project No: P001

แพลตฟอร์มดิจิทัลสำหรับการตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและมะเร็งช่องปาก  
Digital Platform for Detecting and Analyzing Oral Potentially Malignant Disorders and Oral Cancer

**Members**  
ชาญชัย โยธส (630610726)  
เกวฤทธิ์ สมฤทธิ์ (630610731)

**Committee**  
Assoc. Prof. Dr. ปัทมา วุฒิสารวันนา / Patiwet Wuttisarnwattana (Advisor)  
Assoc. Prof. Dr. ศันสนีย์ เอื้อเฟื้อ630610731 / Sansanee Auephanwiriyaikul  
Asst. Prof. โดม โพธิ์ทวน / Dome Potikanond

**Project Description**  
No Description

## 4. ค้นหา Advisor Information สามารถค้นหาข้อมูลของอาจารย์ที่เกี่ยวข้องกับโครงการได้

### (a) เลือกสาขาวิชาที่ต้องการค้นหา

CMU ENG PROJECT

THANAWIN SAITHONG 640610304 TS

My Project Search Project Advisor Info

### Advisor Directory

Browse and search for advisors across all programs

Total Advisors: 33 (CPE) Computer Engineering Search by name or email...

NAME	EMAIL	PROGRAM
<b>S</b> Assoc. Prof. Dr. Santi Phithakkitnukoon ส.อ.ส. สันติ พิทักษ์กิตติกอน	santi.p@cmu.ac.th	CPE
<b>K</b> Asst. Prof. Dr. Karn Patanukhom อ.อ.ส. คานต์ ปัทนุกหม	karn.patanukhom@cmu.ac.th	CPE
<b>N</b> Asst. Prof. Dr. Navadon Khunlertgit อ.อ.ส. นวอดน คุนลฤทธิ	navadon.k@cmu.ac.th	CPE
<b>P</b> Assoc. Prof. Dr. Patiwet Wuttisarnwattana ส.อ.ส. ปัทมา วุฒิสารวันนา	patiwet.w@cmu.ac.th	CPE
<b>T</b> Assoc. Prof. Dr. Trasapong Thaiupathump ส.อ.ส. ตรีสารพงศ์ ไทยอุปัทม	trasapong.t@cmu.ac.th	CPE
<b>S</b> Dr. Sasin Janpuangtong อ.อ.ส. ศาสิน จันทร์พวงทอง	sasin.ja@cmu.ac.th	CPE
<b>A</b> Assoc. Prof. Dr. Anya Weerapran Apavatjrit อ.อ.ส. อanya วีระปราน อพาวตจรร	anya.a@cmu.ac.th	CPE

### (b) ดูข้อมูลของอาจารย์ว่าเคยเป็นที่ปรึกษาหรือกรรมการของโครงการไหนบ้าง





## Create Project


### Project Details

Course *	Section	Semester	Academic Year
<input type="text" value="201492"/>	<input type="text" value="002"/>	<input type="text" value="2"/>	<input type="text" value="2567"/>

Project Title (EN) \*

Project Title (TH) \*

Abstract



### Team Details

Students \*

Advisor \*



Co-Advisor



Committee Members



External Committee Members



### Uploads

**Report**

Upload File

Maximum file size: 25MB

No file chosen

**Poster**

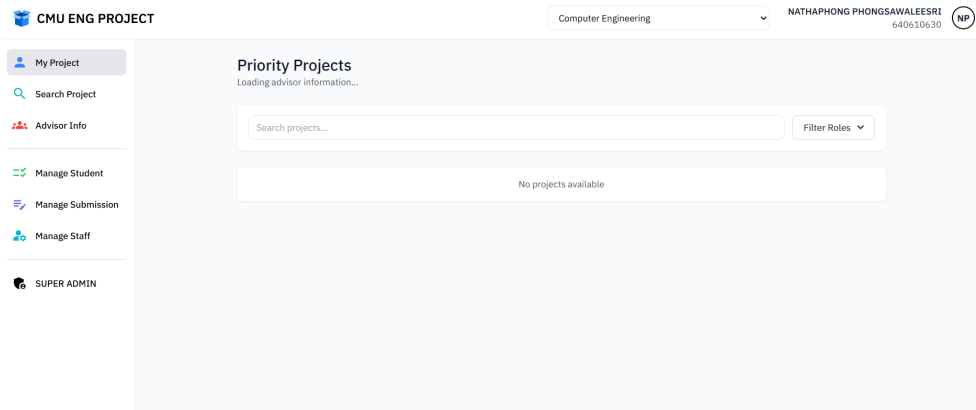
Upload File

Maximum file size: 25MB

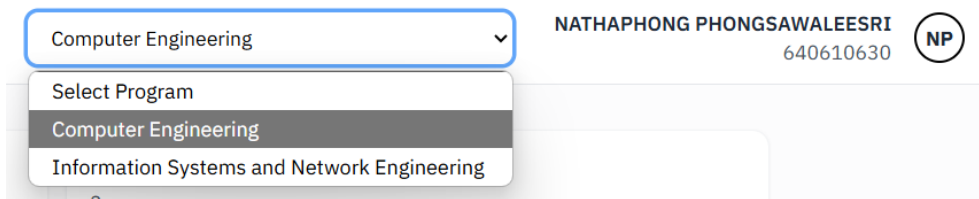
No file chosen

Submit

## • การใช้งานในส่วนของผู้หน้าที่ประจำสาขา **Program Admin**



### 1. เลือก Program ที่ต้องการจัดการโครงการ



### 2. จัดการนักศึกษา (Manage Student):

เจ้าหน้าที่สามารถจัดการข้อมูลนักศึกษาและสิทธิ์ในการสร้างโครงการได้ โดยคลิกที่ปุ่ม "Manage Student" เพื่อเพิ่ม หรือตรวจสอบข้อมูลนักศึกษามีสิทธิ์ในการสร้างโครงการในปีการศึกษาและภาคการศึกษานั้นๆ


- (a) **เพิ่มข้อมูลนักศึกษา** เจ้าหน้าที่สามารถเพิ่มรายชื่อนักศึกษาเข้าสู่ระบบได้ โดยการนำเข้าข้อมูลจากไฟล์ Excel ที่ได้จากสำนักทะเบียน

### Upload Student

Upload a list of students to grant them permission to create projects for the current academic year and semester. This allows students to submit their own project details.

**⚠ Please ensure you upload students with the correct Academic Year and Semester. Incorrect academic period may result in students being unable to access the system.**

### Download Roster\_Student\_Template



Click to upload or drag and drop

.xlsx file only (MAX. 800 KB)

**Upload Student List**

รายชื่อนักศึกษาที่จะเป็นทีม ภาคเรียนที่ 2 ปีการศึกษา 2567					
COURSE NO : 261492 โทษขอข้อมูลเมื่อ 9 ม.ค. 2568 11:32:03					
TITLE :					
ที่	SECLC	SECLAB	รหัสนักศึกษา	ชื่อ-นามสกุล	
1	000	001	640610622	ศุภพร	ฟ้าหทัย ฟ้าหทัย.ckkpop_u@cmu.ac.th
2	000	001	640610632	ณัฐ	ฟ้าหทัย ฟ้าหทัย.nat_uc@cmu.ac.th
3	000	001	640610651	ปัทมพร	ฟ้าหทัย ฟ้าหทัย.piyaporn_hh@cmu.ac.th
4	000	002	640610304	ธนวันท์	ฟ้าหทัย ฟ้าหทัย.nattaporn_saihong@cmu.ac.th
5	000	002	640610630	ณัฐพร	ฟ้าหทัย ฟ้าหทัย.nattaphong_p@cmu.ac.th
6	000	002	640610653	พิชญพร	ฟ้าหทัย ฟ้าหทัย.pichayorn_hh@cmu.ac.th

รูปที่ ข.1: ตัวอย่างไฟล์ Excel สำหรับนำเข้าข้อมูลนักศึกษา

### (b) จัดการโครงการ: เจ้าหน้าที่สามารถดำเนินการดังนี้

- i. สร้างโครงการเก่า: เพิ่มข้อมูลโครงการที่นักศึกษาเคยทำไว้แล้วในอดีต
- ii. มอบหมายโครงการ: เพิ่มโครงการใหม่และกำหนดให้นักศึกษาดำเนินการ โดยนักศึกษาไม่จำเป็นต้องสร้างโครงการเอง

### Upload Project

Upload projects directly as staff. Use this option to create projects on behalf of students or to import existing/old projects into the system.

⚠ Please verify that the Academic Year and Semester in your project list match the intended academic period. Incorrect values may cause projects to appear in the wrong term.

[Download Roster\\_Project\\_Template](#)



Click to upload or drag and drop

.xlsx file only (MAX. 800 KB)

Upload Project List

COURSE NO : 261492			แบบ Form สร้าง Project สำหรับนักศึกษา					
Title (TH)	Title (EN)	Abstract	SECLAB	Student(s)	ชื่อ - นามสกุล	Committee(s)	Staff Role	
ที่นั่งที่ยังว่างอยู่ seats in the plane	ที่นั่งที่ยังว่างอยู่ seats in the plane		013	620610783	ชวลลิกษณ	แก้วมุล	อาจารย์ที่ปรึกษา	
			013	620610792	ธนดล	ตระกูลชัยน	โสม โพธิ์เกษมเกียรติ สำหรับ รณนิวงศ์	กรรมการ กรรมการ
สำหรับห้องเรียน Smart Clas ในการเข้าถึงโปร	สำหรับห้องเรียน Smart Clas ในการเข้าถึงโปร		801	640610630	ปริญ	เลิศนันทพร	อาจารย์ที่ปรึกษา	
						โสม โพธิ์เกษมเกียรติ ชินวธร อัคราดีลักษณ์	กรรมการ กรรมการ	
การจัดการเอกสาร/management ทางและกำหนด	การจัดการเอกสาร/management ทางและกำหนด		811	620612150	ธนฤต	โยกาส	อาจารย์ที่ปรึกษา	
						ยุทธพงษ์ สมจิต ณัฐพันธ์ พรหมสุข โสม โพธิ์เกษมเกียรติ	กรรมการ กรรมการ กรรมการ	
โสมโพธิ์เกษมเกียรติ	โสมโพธิ์เกษมเกียรติ		001	620610785	บุษผะพงษ์	ปริญ	อาจารย์ที่ปรึกษา	

รูปที่ ข.2: ตัวอย่างไฟล์ Excel สำหรับนำเข้าข้อมูลโครงการ

3. จัดการรูปแบบการส่งโครงการ (Manage Submission): เจ้าหน้าที่สามารถกำหนดรูปแบบการส่งโครงการให้กับนักศึกษาได้

The image shows two parts of a web application interface. The top part is titled "Project Form Configuration" with the subtitle "Configure which fields should be included in the project form". It contains two columns of toggle switches. The first column, "Project Details", includes Title Th, Title En, Abstract Text, Academic Year, Semester, Section Id, and Course Id. The second column, "Members", includes Advisor, Co Advisor, External Committee, Committee, and Student. All toggle switches are currently turned on. A "Save Changes" button is located at the bottom right of this section. The bottom part is titled "Resource Configuration" with the subtitle "Manage uploadable resources for projects". It features a table with columns: ICON, TITLE, TYPE, STATUS, and ACTION. There are two rows: "Report" (File type, Status on) and "Poster" (File type, Status on). Each row has an "Edit" button in the ACTION column. A "+ Add Resource" button is located at the top right of this section.

Project Details	Members
Title Th	Advisor
Title En	Co Advisor
Abstract Text	External Committee
Academic Year	Committee
Semester	Student
Section Id	
Course Id	

ICON	TITLE	TYPE	STATUS	ACTION
	Report	File	<input checked="" type="checkbox"/>	Edit
	Poster	File	<input checked="" type="checkbox"/>	Edit

รูปที่ ข.3: รูปแบบการส่งโครงการ

The image shows a modal window titled "Add Resource" with a close button (X) in the top right corner. It has two main sections. The first section, "Select Icon", displays a grid of icons: CAD, File, Poster, SketchUp, PDF, URL, Picture, PowerPoint, GitHub, and YouTube. The second section, "Or Upload Custom Icon", includes a "Choose File" button, a text input field for "Title" (placeholder: "Enter resource title"), and a "Resource Type" dropdown menu currently set to "File Upload". At the bottom are "Cancel" and "Add Resource" buttons. In the background, a blurred view of the "Resource Configuration" table is visible.

รูปที่ ข.4: กำหนดสิ่งที่นักศึกษาต้องส่งในโครงการของแต่ละสาขา เช่น รายงาน, โค้ด, ฯลฯ

4. จัดการคณาจารย์ในสาขา (Manage Staff): เจ้าหน้าที่ที่สามารถจัดการข้อมูลของอาจารย์ในสาขาได้

[Download Roster\\_Staff\\_Template](#)

Manage Staff

Add StaffImport Excel

Filter by name or email

ID	NAME	EMAIL	ACTIONS	STATUS
1	<div>S</div> Assoc. Prof. Dr. Santi Phithakkitnukoon สท.อศ. สันติ พิทักษ์นุกoon	santi.p@cmu.ac.th	Edit	<input checked="" type="checkbox"/>
2	<div>K</div> Asst. Prof. Dr. Karn Patanukhom ผศ.ดร. Karn ปัทนุกhom	karn.patanukhom@cmu.ac.th	Edit	<input checked="" type="checkbox"/>
3	<div>N</div> Asst. Prof. Dr. Navadon Khunlertgit ผศ.ดร. นวอดน ขุนเลิศgit	navadon.k@cmu.ac.th	Edit	<input checked="" type="checkbox"/>
4	<div>P</div> Assoc. Prof. Dr. Patiwet Wuttisarnwattana สท.อศ. ปติwet วุตติสรณวattana	patiwet.w@cmu.ac.th	Edit	<input checked="" type="checkbox"/>
5	<div>T</div> Assoc. Prof. Dr. Trasapong Thaiupathump สท.อศ. ธรสพong ไทยูปathump	trasapong.t@cmu.ac.th	Edit	<input checked="" type="checkbox"/>

## Add New Staff

Position (EN)

First Name (EN)

Last Name (EN)

ตำแหน่ง (TH)

ชื่อจริง (TH)

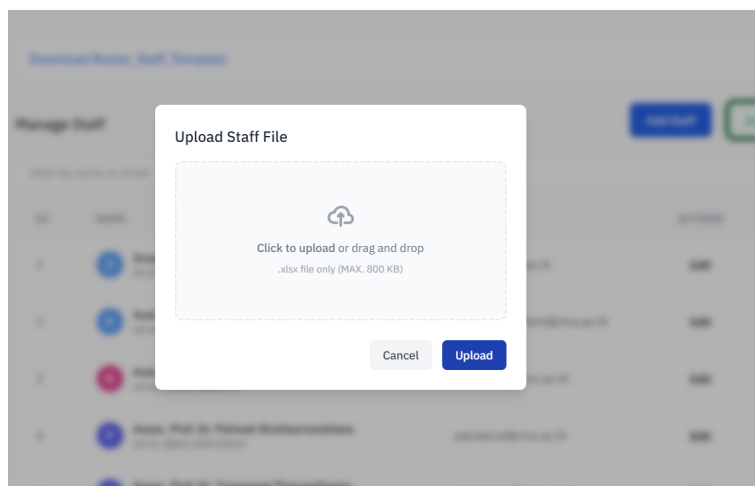
นามสกุล (TH)

Email

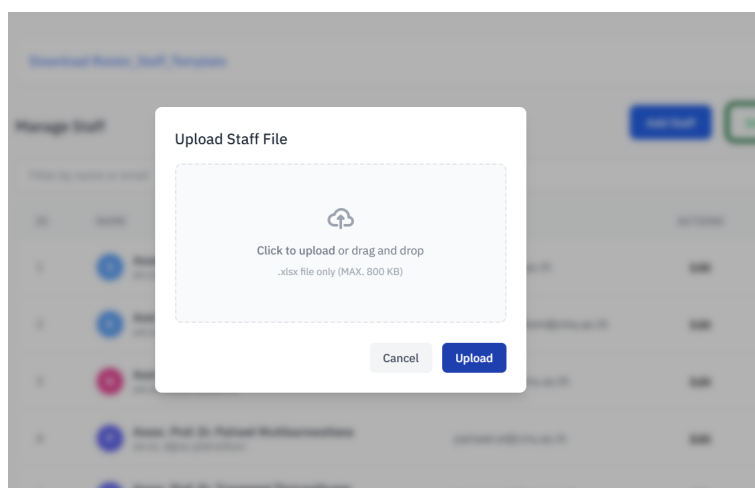
Active Status ☒

Cancel

Save



รูปที่ ข.6: เพิ่มข้อมูลอาจารย์ โดยการนำเข้าข้อมูลจากไฟล์ Excel ที่ได้จากสำนักทะเบียน



รูปที่ ข.5: ตัวอย่างไฟล์ Excel สำหรับนำเข้าข้อมูลอาจารย์

แบบฟอร์มสร้าง Staff							
Prefix (TH)	Prefix (EN)	ชื่อ-นามสกุล (TH)		ชื่อ-นามสกุล (EN)		Email (required)	IsResigned
รศ.ดร.	Assoc. Prof. Dr.	สันติ	พิทักษ์วิบูลย์	Santi	Phithakwiboon	santi.p@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	กานต์	ปัทมทอง	Karn	Patanukhom	karn.patanukhom@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	นาวอน	คุณเลิศกิจ	Navadon	Khunlertgit	navadon.k@cmu.ac.th	FALSE
รศ.ดร.	Assoc. Prof. Dr.	ปัทม	วุฒิสารวัฒนา	Patwet	Wuttisarnwattana	patwet.w@cmu.ac.th	FALSE
รศ.ดร.	Assoc. Prof. Dr.	ตราพงษ์	ไทยอุปนิม	Trasapong	Thaiupathump	trasapong.t@cmu.ac.th	FALSE
อ.ดร.	Dr.	ศสิน	จันทรวงทอง	Sasin	Janpuangtong	sasin.ja@cmu.ac.th	FALSE
รศ.ดร.	Assoc. Prof. Dr.	อัญญา	ฉาภาวิฑูรย์ วีระประพันธ์	Anya	Weerapapan Apavatjrit	anya.a@cmu.ac.th	FALSE
รศ.ดร.	Assoc. Prof. Dr.	จิตรพงศ์	นางวชิร	Juggapong	Natwichai	juggapong.n@cmu.ac.th	FALSE
รศ.ดร.	Assoc. Prof. Dr.	นริศรา	เนรมิตเดชาดี	Kenneth	Cosh	kenneth.c@cmu.ac.th	FALSE
รศ.ดร.	Assoc. Prof. Dr.	ศักดิ์ชัย	ระวีวงศ์	Narissara	Elamkanitchat	narissara.e@cmu.ac.th	FALSE
รศ.ดร.	Assoc. Prof. Dr.	สรวิชัย	กมลบุตร	Sakgait	Ramingwong	sakgait.ramingwong@cmu.ac.th	FALSE
รศ.ดร.	Assoc. Prof. Dr.	ศันชัย	เอื้ออำนวย	Sarnawat	Kantabutra	sarnawat.k@cmu.ac.th	FALSE
รศ.	Asst. Prof.	โดม	โพธิ์เกษม	Sansanee	Auephanwityakul	sansanee.s@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	กมล	วชิษฐ์	Dome	Potikanond	dome.potikanond@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	เกษมสิทธิ์	ดิษฐ์	Kampol	Woradit	kampol.w@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	เสนา	ระวีวงศ์	Kasemsit	Teeyapan	kasemsit.t@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	ณัฐ	พรหมบุตร	Lachana	Ramingwong	lachana.ramingwong@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	ณัฐ	ณัฐ	Nathanan	Promsak	nathanan.p@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	พชร	ณัฐ	Pakorn	Champrasert	pakorn.c@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	ธนทัต	ธนทัต	Pruet	Boonma	pruet.b@cmu.ac.th	FALSE
รศ.ดร.	Asst. Prof. Dr.	ธนทัต	ธนทัต	Thanatip	Chankong	thanatip.ch@cmu.ac.th	FALSE

รูปที่ ข.7: ตัวอย่างไฟล์ Excel สำหรับนำเข้าข้อมูลอาจารย์

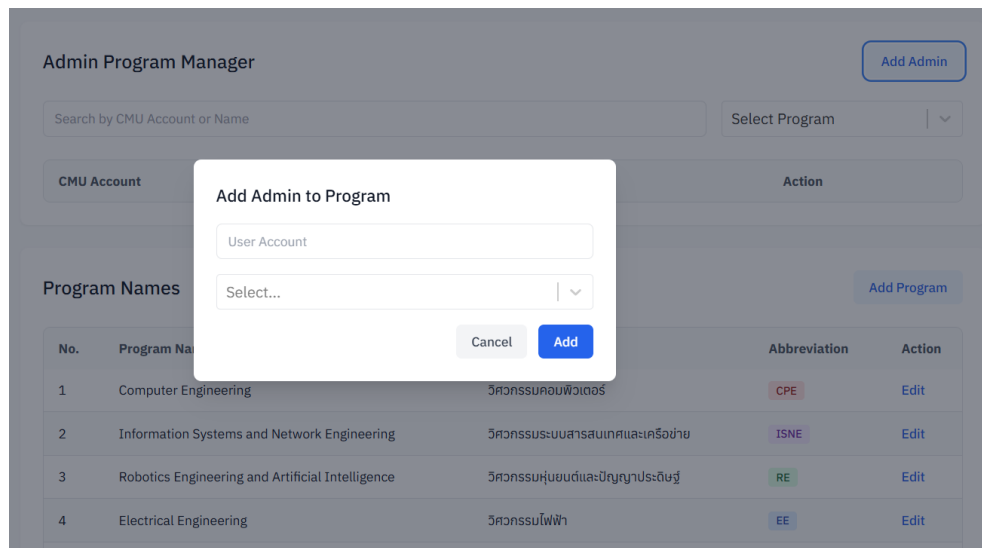
- **ผู้ดูแลระบบแพลตฟอร์ม (Super Admin):** สามารถจัดการข้อมูลของสาขาและโครงการทั้งหมดในระบบ รวมถึงจัดการสิทธิ์ให้กับเจ้าหน้าที่ประจำสาขา

No.	Program Name En	Program Name Th	Abbreviation	Action
1	Computer Engineering	วิศวกรรมคอมพิวเตอร์	CPE	Edit
2	Information Systems and Network Engineering	วิศวกรรมระบบสารสนเทศและเครือข่าย	ISNE	Edit
3	Robotics Engineering and Artificial Intelligence	วิศวกรรมหุ่นยนต์และปัญญาประดิษฐ์	RE	Edit
4	Electrical Engineering	วิศวกรรมไฟฟ้า	EE	Edit
5	Mechanical Engineering	วิศวกรรมเครื่องกล	ME	Edit

No.	Program Name En	Program Name Th	Abbreviation	Action
6	Civil Engineering	วิศวกรรมโยธา	CE	Edit
7	Industrial Engineering	วิศวกรรมอุตสาหกรรม	IE	Edit
8	Integrated Engineering	วิศวกรรมบูรณาการ	INE	Edit
9	Environmental Engineering	วิศวกรรมสิ่งแวดล้อม	ENVI	Edit

รูปที่ ข.8: จัดการข้อมูลสาขา





รูปที่ ข.9: เพิ่มเจ้าหน้าที่ประจำสาขา