

โครงการเลขที่ วศ.คพ. P008-2/2567

เรื่อง

สกอร์โอบีอีพลัส: ระบบจัดการข้อมูลสารสนเทศสำหรับการศึกษาที่เน้นผลลัพธ์

โดย

นางสาว รนพร ชาญชนะโยยิน รหัส 640610638

นางสาว วรพิชชา เมืองยศ รหัส 640610666

นาย สวิช จาเริกพูนผล รหัส 640610672

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ปีการศึกษา 2567

**PROJECT No. CPE P008-2/2567**

**ScoreOBE+: MIS for Outcome-based Education**

**Thanaporn Chanchanayothin 640610638**

**Worapitcha Muangyot 640610666**

**Sawit Charuekpoonpol 640610672**

**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 : សកอร์ໂອබີເພລັສ: ระบบจัดการข้อมูลสารสนเทศสำหรับการศึกษาที่เน้นผลลัพธ์  
โดย : ScoreOBE+: MIS for Outcome-based Education  
นางสาว รนพร ชาญชนะโยธิน รหัส 640610638  
นางสาว วรพิชชา เมืองยศ รหัส 640610666  
นาย สวิช จาเร็กพูนผล รหัส 640610672  
ภาควิชา : วิศวกรรมคอมพิวเตอร์  
อาจารย์ที่ปรึกษา : ผศ. โอดม โพธิ์กานนท์  
ปริญญา : วิศวกรรมศาสตรบัณฑิต  
สาขา : วิศวกรรมคอมพิวเตอร์  
ปีการศึกษา : 2567

---

ภาควิชา/วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชา/วิศวกรรมคอมพิวเตอร์  
(รศ.ดร. สันติ พิทักษ์กิจนุภร)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ  
(ผศ. โอดม โพธิ์กานนท์)

..... กรรมการ  
(ผศ.ดร. นวดนัย คุณเลิศกิจ)

..... กรรมการ  
(ผศ.ดร. ธนาทิพย์ จันทร์คง)

หัวข้อโครงการ	: สกอร์โอปีอีพลัส: ระบบจัดการข้อมูลสารสนเทศสำหรับการศึกษาที่เน้นผลลัพธ์
	: ScoreOBE+: MIS for Outcome-based Education
โดย	: นางสาว ชนพร ชาญชนะโยธิน รหัส 640610638 นางสาว วรพิชชา เมืองยศ รหัส 640610666 นาย สวิช จาเร็กพูนผล รหัส 640610672
ภาควิชา	: วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	: ผศ. โภม โพธิ์กานนท์
ปริญญา	: วิศวกรรมศาสตรบัณฑิต
สาขา	: วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	: 2567

---

## บทคัดย่อ

สกอร์โอปีอีพลัส เป็นระบบบริหารจัดการข้อมูลด้านผลลัพธ์อีทั้งการศึกษาที่พัฒนาขึ้นสำหรับคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ระบบนี้ออกแบบมาเพื่อช่วยให้ผู้สอนสามารถประกาศคะแนนสอบและงานที่มีอิทธิพลต่อการเรียนรู้ พร้อมทั้งสนับสนุนการวิเคราะห์ผลลัพธ์ของนักศึกษาในรูปแบบที่สอดคล้องกับแนวทางของ กรอบมาตรฐานคุณวุฒิระดับอุดมศึกษาแห่งชาติ (TQF: Thailand Qualifications Framework) และแนวคิดการจัดการศึกษาโดยเน้นผลลัพธ์ (Outcome-Based Education: OBE)

ปัญหาสำคัญของการบูรณาการจัดทำเอกสาร TQF คือการรวมและวิเคราะห์ข้อมูลผลการเรียนรู้ของนักศึกษาให้เป็นไปตามตัวชี้วัดที่กำหนด ซึ่งมักเป็นงานที่ซับซ้อนและใช้เวลานาน ผู้สอนต้องดึงข้อมูลคะแนนมาสรุปผลและประเมินว่าผู้เรียนสามารถบรรลุผลลัพธ์การเรียนรู้ตามที่หลักสูตรกำหนดหรือไม่ อย่างไรก็ตาม การดำเนินการเหล่านี้มักขาดเครื่องมือที่ช่วยในการประมวลผลอย่างเป็นระบบ ทำให้เกิดภาระงานเอกสารที่มากขึ้น และอาจนำไปสู่ความคลาดเคลื่อนในการสรุปผล

สกอร์โอปีอีพลัส ถูกพัฒนาขึ้นเพื่อแก้ปัญหานี้ โดยช่วยให้ผู้สอนสามารถอัปโหลดคะแนนผ่านไฟล์ .xlsx/.csv พร้อมการคำนวณค่าสถิติที่เกี่ยวข้อง เช่น ค่าเฉลี่ย ค่ามัธยฐาน และส่วนเบี่ยงเบนมาตรฐาน ระบบเชื่อมโยงข้อมูลคะแนนกับผลลัพธ์การเรียนรู้ใน TQF โดยอัตโนมัติ ลดภาระงานของผู้สอน และช่วยให้การสรุปผลลัพธ์เป็นไปอย่างแม่นยำและรวดเร็ว

นอกจากนี้ ระบบยังช่วยให้การประกาศคะแนนเป็นไปอย่างเป็นระบบ นักศึกษาสามารถเข้าถึงผลการเรียนของตนเอง ผ่านอินเทอร์เฟซที่เข้าใจง่าย พร้อมข้อมูลผลลัพธ์การเรียนรู้ที่เกี่ยวข้อง ส่งเสริมให้ผู้สอนและนักศึกษามองเห็นภาพรวมของการเรียนรู้ได้ชัดเจนขึ้น

ผู้จัดทำหวังว่า สกอร์โอปีอีพลัส จะเป็นเครื่องมือสำคัญที่ช่วยปรับปรุงคุณภาพการศึกษาให้สอดคล้องกับแนวทางของ OBE และ TQF และสามารถขยายการใช้งานไปยังคณะหรือหน่วยงานอื่น ๆ ได้ในอนาคต

## ABSTRACT

**ScoreOBE+** is an educational outcome management system developed for the Faculty of Engineering, Chiang Mai University. This system is designed to assist instructors in conveniently announcing exam and assignment scores, while also supporting student performance analysis in accordance with the Thailand Qualifications Framework (TQF) and the principles of Outcome-Based Education (OBE).

A major challenge in preparing TQF documents is the collection and analysis of student learning outcomes based on predefined indicators. This process is often complex and time-consuming, requiring instructors to extract score data, summarize results, and evaluate whether students have achieved the expected learning outcomes. However, the lack of systematic tools for data processing increases the workload and can lead to inaccuracies in outcome assessments.

**ScoreOBE+** was developed to address these issues by enabling instructors to upload scores via xlsx/csv files, automatically calculating key statistics such as mean, median, and standard deviation. The system links score data to TQF learning outcomes automatically, reducing instructors' workload and ensuring accurate and efficient outcome reporting.

Moreover, the system streamlines the score announcement process, allowing students to access their results through a user-friendly interface that includes relevant learning outcome information. This feature enhances transparency and helps both instructors and students gain a clearer understanding of the learning process.

The developers hope that **ScoreOBE+** will serve as a vital tool for improving education quality in alignment with OBE and TQF standards, with the potential for future expansion to other faculties or institutions.

## กิตติกรรมประกาศ

โครงการ **ScoreOBE+** ไม่อาจสำเร็จลุล่วงได้ หากปราศจากแรงสนับสนุน คำแนะนำ และความทุ่มเท จากหลายฝ่าย พากเราขอถือโอกาสสนับสนุน แสดงความขอบคุณจากใจจริงต่อทุกภาคส่วนที่มีส่วนร่วมในการผลักดัน ให้โครงการนี้เกิดขึ้นและพัฒนาไปสู่จุดที่สามารถนำไปใช้งานจริงได้

ก่อนอื่น พากเราขอขอบคุณ ผศ. โอม โพธิ์กานนท์ อาจารย์ที่ปรึกษาโครงการ ซึ่งเป็นสาหหลักสำคัญของ การพัฒนา **ScoreOBE+** ไม่เพียงให้คำแนะนำด้านวิชาการและเทคนิคอย่างต่อเนื่อง แต่ยังคอยสนับสนุน และชี้แนะทิศทางที่เหมาะสมในทุกช่วงเวลาของการพัฒนาโครงการ เสียสละเวลาเพื่อประชุม ให้คำปรึกษา และช่วยให้พากเราเข้าใจแนวคิดสำคัญที่จำเป็นต่อการทำให้ระบบมีประสิทธิภาพสูงสุด

นอกจากนี้ พากเราขอขอบคุณ ผศ.ดร. นวนดย คุณเลิศกิจ และ ผศ.ดร. ธนาทิพย์ จันทร์คง อาจารย์กรรมการสอบโครงการ ที่สละเวลาในทุกสัปดาห์เพื่อตรวจสอบ ให้ข้อเสนอแนะ และแนะนำแนวทาง ในการปรับปรุงระบบให้มีคุณภาพและความสมบูรณ์ยิ่งขึ้น คำแนะนำของทั้งสองไม่เพียงช่วยพัฒนาระบบที่ ตอบโจทย์ทางวิชาการ แต่ยังช่วยให้พากเราได้เรียนรู้และติบโตจากประสบการณ์การทำงานและลงพื้นที่จริง มากไปกว่านั้น พากเราขอขอบคุณ อ.ดร. ชินวัตร อิศราดิสัยกุล สำหรับคำแนะนำเกี่ยวกับแหล่งอ้างอิงในการสร้างกราฟการกระจายตัวของคะแนน grader ซึ่งอ้างอิงจากแนวคิดของ Andrew C. Myers (Cornell University) ที่ใช้หลักการ *a posteriori* ในการสร้างแผนภาพการกระจายคะแนน

พากเราขอขอบคุณ ภาควิชาศิวกรรมคอมพิวเตอร์ คณะศิวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ที่ ให้การสนับสนุนด้านทรัพยากร เทคโนโลยี และองค์ความรู้ ซึ่งเป็นฐานสำคัญในการพัฒนาโครงการ รวม ถึงนักศึกษาและอาจารย์ทุกท่านที่ให้ข้อเสนอแนะอันมีค่า การสนับสนุนความคิดเห็นของทุกท่านช่วยให้พากเรา สามารถปรับปรุงระบบให้ตอบสนองความต้องการของผู้ใช้งานได้อย่างแท้จริง

และเนื่องอสิ่งอื่นใด พากเราขอขอบคุณสมาชิกทีมพัฒนา ทุกคน ที่ร่วมแรงร่วมใจ ทุ่มเททั้งเวลา ความคิด และพลังงานอย่างเต็มที่เพื่อให้โครงการนี้เกิดขึ้นได้จริง พากเราต่างผ่านอุปสรรค ความท้าทาย และช่วงเวลาที่ เหนื่อยล้า แต่ทุกคนยังคงมุ่งมั่นที่จะผลักดันให้ **ScoreOBE+** เป็นระบบที่สามารถช่วยสนับสนุนกระบวนการ ประเมินผลการเรียนรู้ให้มีประสิทธิภาพมากที่สุด

โครงการนี้ไม่ได้เป็นเพียงแค่ซอฟต์แวร์ที่ช่วยอำนวยความสะดวกและจัดการข้อมูลสารสนเทศสำหรับการ ศึกษาที่เน้นผลลัพธ์เท่านั้น แต่ยังเป็นผลลัพธ์ของความพยายามร่วมกันของทีมพัฒนา คณาจารย์ และผู้เกี่ยวข้องทุกฝ่ายที่มุ่งเน้นคุณค่าและประโยชน์ของระบบนี้ พากเราหวังเป็นอย่างยิ่งว่า **ScoreOBE+** จะสามารถนำไปใช้ให้เกิดประโยชน์ต่อคณะศิวกรรมศาสตร์ และขยายขอบเขตการใช้งานจากระดับคณะไปสู่ระดับมหาวิทยาลัย จนกระทั่งสามารถพัฒนาให้เป็นมาตรฐานที่ใช้งานได้ในระดับอุดมศึกษาของมหาวิทยาลัย อื่น ๆ เพื่อช่วยให้นักศึกษาและบุคลากรทางการศึกษาสามารถใช้ข้อมูลในการประเมินผลการเรียนรู้ได้อย่าง มีประสิทธิภาพ และนำไปสู่การพัฒนาคุณภาพการศึกษาต่อไปในอนาคต

นางสาว รัชนพร ชาญชนาญอิน

นางสาว วรพิชชา เมืองยศ

นาย สวิช จาเริกพูนผล

22 มีนาคม 2568

# สารบัญ

บทคัดย่อ . . . . .	ข
Abstract . . . . .	ค
กิตติกรรมประกาศ . . . . .	ง
สารบัญ . . . . .	จ
สารบัญรูป . . . . .	ช
สารบัญตาราง . . . . .	ญ
<b>1 บทนำ . . . . .</b>	<b>1</b>
1.1 ที่มาของโครงการ . . . . .	1
1.2 วัตถุประสงค์ของโครงการ . . . . .	2
1.3 ขอบเขตของโครงการ . . . . .	2
1.3.1 ขอบเขตด้านผู้ใช้งาน . . . . .	2
1.3.2 ขอบเขตด้านซอฟต์แวร์ . . . . .	3
1.4 ประโยชน์ที่ได้รับ . . . . .	4
1.5 เทคโนโลยีและเครื่องมือที่ใช้ . . . . .	4
1.5.1 เทคโนโลยีด้านハードแวร์ . . . . .	4
1.5.2 เทคโนโลยีด้านซอฟต์แวร์ . . . . .	5
1.6 แผนการดำเนินงาน . . . . .	6
1.7 บทบาทและความรับผิดชอบ . . . . .	7
1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม . . . . .	7
<b>2 ทฤษฎีที่เกี่ยวข้อง . . . . .</b>	<b>9</b>
2.1 ความหมายของ OBE . . . . .	9
2.2 แพลตฟอร์มที่คล้ายคลึงกัน . . . . .	9
2.2.1 CPE QA . . . . .	9
2.2.2 CMU Score Announcement . . . . .	10
2.2.3 CMU OBE . . . . .	10
2.3 User Experience and User Interface . . . . .	12
2.3.1 User-Centered Design (UCD) . . . . .	12
2.3.2 Jakob's Law . . . . .	12
2.3.3 Gestalt Principles . . . . .	12
2.3.4 Interaction Design Principle . . . . .	13
2.3.5 Design Thinking . . . . .	14
2.3.6 Usability Testing . . . . .	15
2.3.7 Feedback Capture Grid . . . . .	16
2.4 Platform Development . . . . .	17
2.4.1 MVC . . . . .	17
2.4.2 Rest API . . . . .	18
2.4.3 Role-Based Access Control (RBAC) . . . . .	18
2.4.4 JSON Web Token (JWT) . . . . .	20
2.4.5 Containerization (Docker) . . . . .	22
2.4.6 Separation of Concerns (SoC) . . . . .	23
2.4.7 Modular Architecture . . . . .	24
2.5 Database . . . . .	25
2.6 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ . . . . .	25

2.7 ความรู้นักศึกษาที่จะนำไปใช้หรือบูรณาการในโครงการ . . . . .	25
<b>3 โครงสร้างและขั้นตอนการทำงาน</b>	<b>26</b>
3.1 สถาปัตยกรรมระบบ . . . . .	26
3.1.1 Authentication . . . . .	26
3.1.2 Frontend . . . . .	27
3.1.3 Backend . . . . .	28
3.1.4 Database . . . . .	29
3.1.5 NGINX และ Docker . . . . .	30
3.2 Database Design . . . . .	30
3.2.1 โครงสร้างของฐานข้อมูลใน MongoDB . . . . .	30
3.2.2 โครงสร้างออกแบบเจ็กต์ในระบบฐานข้อมูล . . . . .	33
3.2.3 ตัวอย่างโครงสร้างข้อมูล . . . . .	34
3.3 การคำนวนคะแนน CLO (Course Learning Outcomes) . . . . .	36
3.3.1 ขั้นตอนการเตรียมการก่อนการคำนวน . . . . .	36
3.3.2 กระบวนการคำนวนคะแนน CLO . . . . .	38
3.3.3 การคำนวนคะแนน CLO (Course Learning Outcomes) . . . . .	39
3.3.4 การคำนวนคะแนน PLO (Program Learning Outcomes) . . . . .	42
3.4 การคำนวนคะแนน PLO . . . . .	42
3.4.1 ตัวอย่างการคำนวน . . . . .	42
3.5 API Endpoint . . . . .	43
<b>4 การทดลองและผลลัพธ์</b>	<b>44</b>
4.1 การทดสอบความถูกต้องของข้อมูลหลักสูตรและระบบวิชา . . . . .	44
4.2 การทดสอบการนำเข้าค่า . . . . .	44
4.2.1 การนำเข้าไฟล์ CSV/XLSX ตามรูปแบบที่กำหนด . . . . .	44
4.2.2 การตรวจสอบความถูกต้องของข้อมูลที่นำเข้า . . . . .	44
4.3 การประเมินผลและวิเคราะห์คะแนน . . . . .	44
4.3.1 การคำนวนสถิติอัตโนมัติ . . . . .	45
4.3.2 การวิเคราะห์การจับคู่ค่าคะแนนกับ TQF 3 และ TQF 5 . . . . .	45
4.3.3 การแสดงผลคะแนนของนักศึกษาในรูปแบบการกระจายคะแนน . . . . .	45
4.4 การทดสอบการใช้งานจากผู้ใช้จริง . . . . .	45
4.4.1 วัตถุประสงค์ของการทดสอบ . . . . .	45
4.4.2 กลุ่มตัวอย่างและวิธีการทดสอบ . . . . .	46
4.4.3 ผลการทดสอบและการวิเคราะห์ข้อมูล . . . . .	47
<b>5 บทสรุปและข้อเสนอแนะ</b>	<b>50</b>
5.1 ข้อจำกัดของระบบ . . . . .	50
5.2 สรุปผล . . . . .	50
5.3 ปัญหาที่พบและแนวทางการแก้ไข . . . . .	51
5.4 ข้อเสนอแนะและแนวทางการพัฒนาต่อ . . . . .	51
5.5 สรุปผล . . . . .	52
<b>บรรณานุกรม</b>	<b>53</b>

<b>ก คู่มือการใช้งานระบบ</b>	<b>57</b>
ก.1 ผู้สอน (Instructor) . . . . .	57
ก.1.1 การจัดทำเอกสารประกอบการเรียนการสอน . . . . .	57
ก.1.2 การจัดการระบบวิชาและการนำเข้าค่าແນน . . . . .	57
ก.1.3 การจัดการคะแนนและการประมวลผลการเรียน . . . . .	64
ก.1.4 การจัดการเอกสาร TQF3 และ TQF5 . . . . .	66
ก.1.5 ข้อจำกัดและเงื่อนไขการใช้งาน . . . . .	70
ก.2 ผู้ดูแลหลักสูตร (Curriculum Administrator) . . . . .	70
ก.2.1 การจัดการเอกสาร TQF3 และ TQF5 . . . . .	70
ก.2.2 การติดตามสถานะเอกสาร TQF3 และ TQF5 . . . . .	71
ก.2.3 การจัดการระบบวิชา . . . . .	71
ก.2.4 การสร้างชุด PLO . . . . .	72
ก.2.5 การเข้มข้อ PLO Require กับ CLO ในแต่ละระบบวิชา . . . . .	72
ก.2.6 การวิเคราะห์การเข้มข้อ CLO กับ PLO . . . . .	73
ก.2.7 การติดตามผลการเรียนรู้ของ PLO . . . . .	73
ก.2.8 การนำออกคะแนน PLO และการจัดทำเอกสาร TQF7 . . . . .	75
ก.2.9 ข้อจำกัดและเงื่อนไขการใช้งาน . . . . .	75
ก.3 ผู้ดูแลระบบ (System Administrator) . . . . .	76
ก.3.1 สิทธิ์การเข้าถึงของผู้ดูแลระบบ . . . . .	76
ก.3.2 การกำหนดผู้ดูแลระบบ . . . . .	76
ก.3.3 การจัดการภาคการศึกษา . . . . .	76
ก.3.4 การจัดการผู้ดูแลหลักสูตร . . . . .	77
ก.3.5 การจัดการหลักสูตร . . . . .	77
ก.3.6 การจัดการระบบวิชา . . . . .	78
ก.3.7 ข้อจำกัดและเงื่อนไขการใช้งาน . . . . .	78
ก.4 นักศึกษา (Student) . . . . .	79
ก.4.1 การดูค่าແນนและการวิเคราะห์ทางสถิติ . . . . .	79
ก.4.2 การดูผลการเรียนรู้ของระบบวิชา (Course Learning Outcome, CLO) . .	80
ก.4.3 การค้นหาข้อมูลโครงสร้างรายวิชา (Course Syllabus) . . . . .	81
ก.4.4 การดูผลการเรียนรู้ระดับหลักสูตร (Program Learning Outcome, PLO) . .	82
ก.4.5 ข้อจำกัดและเงื่อนไขการเข้าถึงข้อมูล . . . . .	83

## สารบัญรูป

2.1 CMU OBE Create Gradebook . . . . .	11
2.2 CMU OBE Create Score . . . . .	11
2.3 CMU OBE Create TQF3 . . . . .	11
2.4 User-Centered Design (UCD) . . . . .	12
2.5 MVC . . . . .	18
3.1 System Diagram . . . . .	26
3.2 Assessment Mapping . . . . .	36
3.3 Map Question in Assessment with Course CLOs . . . . .	37
3.4 Set Range Assessment Tool . . . . .	38
3.5 Curriculum Mapping . . . . .	42
ก.1 Add Course Step 1 . . . . .	58
ก.2 Add Course Step 2 . . . . .	59
ก.3 Add Course Step 3 . . . . .	60
ก.4 Add Course Step 4 . . . . .	60
ก.5 Add Course Step 5 . . . . .	61
ก.6 การนำเข้ารายชื่อนักศึกษา . . . . .	62
ก.7 การนำเข้าค่าแนน . . . . .	62
ก.8 การนำเข้ารายชื่อนักศึกษา . . . . .	63
ก.9 การนำอ่าอกค่าแนนที่เคยอปป์โหลด . . . . .	64
ก.10 การแก้ไขค่าแนนนักศึกษา . . . . .	64
ก.11 การลบค่าแนนงานมอบหมาย . . . . .	65
ก.12 การประกาศค่าแนนงานมอบหมาย . . . . .	66
ก.13 TQF3 Management . . . . .	66
ก.14 TQF3 Management . . . . .	67
ก.15 TQF3 Management . . . . .	67
ก.16 TQF3 Management . . . . .	68
ก.17 TQF3 Management . . . . .	68
ก.18 TQF3 Management . . . . .	69
ก.19 TQF3 Management . . . . .	69
ก.20 TQF3 Management . . . . .	70
ก.21 TQF5 Management . . . . .	71
ก.22 Status TQF . . . . .	71
ก.23 Course Management . . . . .	72
ก.24 Create PLO collection . . . . .	72
ก.25 PLO Mapping . . . . .	73
ก.26 CLO Overview . . . . .	73
ก.27 PLO Term View . . . . .	74
ก.28 PLO Year View . . . . .	74
ก.29 PLO Curriculum View . . . . .	75
ก.30 Export PLO . . . . .	75
ก.31 Admin Management . . . . .	76
ก.32 Semester Management . . . . .	77
ก.33 Curriculum Admin Management . . . . .	77

n.34 Curriculum Management . . . . .	78
n.35 Course Management . . . . .	78
n.36 statistic . . . . .	79
n.37 Histogram . . . . .	80
n.38 Distribution . . . . .	80
n.39 CLOStu . . . . .	81
n.40 CourseSylla . . . . .	81
n.41 SearchCourseSylla . . . . .	82
n.42 StuPLOs . . . . .	82

## สารบัญตาราง

2.1 การกำหนดสิทธิ์การเข้าถึงตามบทบาทใน ScoreOBE+ . . . . .	19
3.1 คะแนนดิบของนักศึกษา . . . . .	40
3.2 คะแนนรวมตามวิธีการประเมิน . . . . .	40
3.3 เกณฑ์การแปลงคะแนนเป็นมาตรฐาน 0-4 . . . . .	40
3.4 คะแนนมาตรฐาน 0-4 ของแต่ละวิธีการประเมิน . . . . .	41
3.5 การคำนวณคะแนน CLO 1 รายบุคคล . . . . .	41
3.6 คะแนน CLO . . . . .	42
3.7 สรุปคะแนน PLO . . . . .	43

## บทที่ 1

### บทนำ

#### 1.1 ที่มาของโครงการ

การศึกษาในระบบ **Outcome-Based Education (OBE)** มุ่งเน้นการพัฒนาและประเมินผลการเรียนรู้ของนักศึกษา โดยอาศัยการประเมินผลที่สามารถสะท้อนถึงทักษะและความรู้ที่นักศึกษาจะได้รับจากการเรียนในแต่ละหลักสูตร ผลลัพธ์ที่ได้จากการเรียนรู้จะต้องสามารถวัดได้ และสามารถนำไปใช้ในด้านการพัฒนาทักษะในอาชีพและการพัฒนาสังคมอย่างมีประสิทธิภาพ ซึ่งทำให้ OBE กลายเป็นแนวทางการศึกษาที่สำคัญในหลายประเทศ รวมถึงประเทศไทย

ในการดำเนินการตามกรอบ มาตรฐานคุณวุฒิระดับอุดมศึกษาแห่งชาติ (มคอ.) หรือ **TQF (Thai Qualifications Framework)** เพื่อสนับสนุนการประเมินผลการเรียนรู้ตามแนวทาง OBE นั้น มหาวิทยาลัยเชียงใหม่ได้จัดทำเอกสารที่เรียกว่า **มคอ.3 (TQF3)** ซึ่งเป็นแผนการสอนที่อธิบายถึงวัตถุประสงค์การเรียนรู้ที่ต้องการให้ผู้เรียนได้พัฒนาในแต่ละกระบวนการวิชาและหลักสูตร และ **มคอ.5 (TQF5)** ซึ่งเป็นรายงานผลการเรียนการสอนหลังสิ้นสุดภาคการศึกษา โดยในกระบวนการเหล่านี้ อาจารย์ต้องทำการบันทึกข้อมูลผลการเรียนรู้ของนักศึกษาในแต่ละขั้นตอน เพื่อให้มั่นใจว่าผลการเรียนรู้ของนักศึกษาตรงตามวัตถุประสงค์ที่กำหนดไว้ใน **TQF** และสอดคล้องกับหลักการของ **OBE**

แม้ว่า **TQF** จะเป็นเครื่องมือสำคัญในการสนับสนุนระบบ **OBE** แต่ในทางปฏิบัติยังพบปัญหาหลายประการในการจัดทำเอกสาร **TQF** ซึ่งส่งผลต่อการประเมินผลการเรียนรู้ของนักศึกษาและการพัฒนาคุณภาพการศึกษา ดังนี้:

1. กระบวนการทำงานที่ยุ่งยากและใช้เวลา many อาจารย์ต้องกรอกข้อมูลลงในเอกสารหลายชุด ซึ่งกินเวลาและเป็นภาระเพิ่มเติมจากการสอน การตรวจสอบและอนุมัติเอกสารใช้เวลานาน ทำให้การสรุปผลการเรียนรู้ล่าช้าและไม่สามารถนำข้อมูลไปใช้ได้อย่างทันท่วงที
2. ความเสี่ยงต่อการสูญหายของข้อมูล การใช้เอกสารในรูปแบบกระดาษส่งผลให้ข้อมูลอาจสูญหายหรือเกิดข้อผิดพลาดระหว่างการส่งต่อเอกสารระหว่างหน่วยงาน ซึ่งส่งผลให้การประเมินผลไม่แม่นยำและไม่สามารถตรวจสอบข้อมูลย้อนหลังได้
3. ความคลาดเคลื่อนของข้อมูลและข้อผิดพลาดในการกรอกเอกสาร อาจารย์ต้องคำนวณและสรุปผลคะแນด้วยตนเอง ทำให้เกิดข้อผิดพลาดได้ง่าย ข้อมูลในรายงาน **TQF5** อาจไม่สะท้อนผลการเรียนรู้ที่แท้จริง
4. ขาดระบบวิเคราะห์ผลการเรียนรู้ที่แม่นยำ ปัจจุบันไม่มีระบบที่สามารถช่วยอาจารย์ในการประเมินผลสัมฤทธิ์ทางการเรียนของนักศึกษาได้โดยอัตโนมัติ ทำให้การวิเคราะห์ผลการเรียนเป็นไปอย่างจำกัดและไม่ได้ผลตามวัตถุประสงค์ที่ตั้งไว้

เพื่อลดปัญหาเหล่านี้ ภาควิชาศิวกรรมคอมพิวเตอร์ จึงได้พัฒนาระบบ **CPE QA** ซึ่งช่วยให้อาจารย์สามารถกรอกข้อมูล **TQF3** และ **TQF5** ได้สะดวกขึ้น อย่างไรก็ตาม ระบบนี้ยังคงมีข้อจำกัด เช่น อาจารย์ยังต้องกรอกข้อมูล **TQF5** ด้วยตนเอง และข้อมูลที่แสดงยังไม่ละเอียดเพียงพอ ซึ่งทำให้ยังไม่สามารถเชื่อมโยงข้อมูลและประเมินผลได้อย่างเต็มที่

นอกจากนี้ ในปีการศึกษา 1/2566 นักศึกษาในกระบวนวิชา **261491 Full-Stack Development** ได้พัฒนา **CMU Score Announcement** ซึ่งเป็นแพลตฟอร์มที่ช่วยในการประกาศคะแนนในกระบวนวิชา ต่างๆ อย่างไรก็ตาม ระบบนี้ไม่ได้ถูกออกแบบให้รองรับการจัดทำ **TQF** โดยตรง ทำให้ยังคงมีข้อจำกัดในการใช้งานในระดับคณะและมหาวิทยาลัย

จากปัญหาดังกล่าว จึงมีความจำเป็นต้องพัฒนาระบบที่สามารถเชื่อมโยงข้อมูลการประเมินผลการเรียนรู้กับระบบ **TQF** อย่างเป็นระบบ ซึ่งจะช่วยลดภาระของอาจารย์ในการกรอกข้อมูลและประเมินผลการเรียนรู้ของนักศึกษา และช่วยให้การจัดทำ **TQF3** และ **TQF5** เป็นไปอย่างมีประสิทธิภาพ พร้อมทั้งเชื่อมโยงข้อมูลการเรียนรู้ในระดับกระบวนวิชาและหลักสูตรกับกรอบ **OBE** ได้อย่างครบถ้วน เพื่อสนับสนุนการพัฒนาคุณภาพการศึกษาอย่างยั่งยืน

## 1.2 วัตถุประสงค์ของโครงการ

- เพื่อช่วยให้นักจัดการงานทั่วไปและอาจารย์สามารถสร้างและส่งออกเอกสารตามกรอบมาตรฐานคุณวุฒิระดับอุดมศึกษาแห่งชาติ (มคอ.) ได้อย่างสะดวกและมีประสิทธิภาพ ระบบจะช่วยให้การจัดทำเอกสาร มคอ.3 (TQF3) และ มคอ.5 (TQF5)
- เพื่อให้การจัดการคะแนนและการวิเคราะห์ข้อมูลทางสถิติเกี่ยวกับผลการเรียนของนักศึกษาทำได้ง่ายและมีประสิทธิภาพ ระบบจะรองรับการนำเข้าคะแนน การคำนวณค่าสถิติที่สำคัญ การสร้างแผนภูมิ Histogram และกราฟการกระจายคะแนน รวมถึงการเผยแพร่ แก้ไข และนำออกคะแนน
- เพื่อให้นักศึกษาสามารถเข้าถึงข้อมูลคะแนนของตนเองและเข้าใจผลลัพธ์การเรียนรู้ที่ได้รับจาก การเรียนในแต่ละวิชา ระบบจะช่วยให้ผู้เรียนสามารถดูคะแนนของตนเอง ร่วมกับการแสดงผลแนวโน้มการกระจายคะแนนและการประเมินผลการเรียนรู้
- เพื่อเชื่อมโยงผลคะแนนของนักศึกษา กับการเรียนรู้ตาม **CLOs (Course Learning Outcomes)** และ **PLOs (Program Learning Outcomes)** ระบบจะช่วยให้การติดตามผลการเรียนรู้ของนักศึกษา โดยการเชื่อมโยงคะแนนที่ได้รับจากการเรียนกับวัตถุประสงค์การเรียนรู้ที่กำหนดไว้ทั้งในระดับวิชา (CLOs) และโปรแกรม (PLOs) เพื่อประเมินผลสัมฤทธิ์การเรียนรู้

## 1.3 ขอบเขตของโครงการ

### 1.3.1 ขอบเขตด้านผู้ใช้งาน

ในโครงการนี้มุ่งเน้นการพัฒนาระบบที่ให้ผู้ใช้งานในกลุ่มต่าง ๆ ดังนี้

- อาจารย์ผู้สอน (Instructor)** อาจารย์ผู้สอนสามารถใช้ระบบในการจัดการข้อมูลการเรียนการสอน เช่น การสร้างและส่งออกเอกสาร มคอ.3 และ มคอ.5 การคำนวณคะแนนพร้อมค่าทางสถิติ และ การสร้างกราฟการกระจายคะแนน โดยมีฟังก์ชันที่ช่วยให้อาจารย์สามารถแก้ไขคะแนนและเผยแพร่ผลการเรียนให้กับนักศึกษา
- นักศึกษา (Student)** นักศึกษาสามารถใช้ระบบในการเข้าถึงข้อมูลผลการเรียนของตนเอง ดูแนวโน้ม การกระจายคะแนน และการวิเคราะห์ผลการเรียนตาม **CLOs (Course Learning Outcomes)**

และ **PLOs (Program Learning Outcomes)** ซึ่งช่วยให้สามารถติดตามความก้าวหน้าในการเรียนรู้และการประเมินผล

- นักจัดการงานทั่วไป (**Department Staff**) นักจัดการงานทั่วไปที่มีบทบาทในการจัดการข้อมูลการศึกษา สามารถใช้ระบบเพื่อช่วยในการจัดการข้อมูล **TQF** ของคณะและภาควิชา รวมถึงการจัดการและประสานงานในการส่งออกเอกสารที่เกี่ยวข้องกับการประเมินผลการเรียนรู้ของนักศึกษา
- กลุ่มผู้ดูแลระบบ (**Admin**) ผู้ดูแลระบบมีบทบาทสำคัญในการรักษาความปลอดภัยของข้อมูลและการควบคุมการเข้าถึงระบบ โดยการจัดการสิทธิ์การเข้าถึงของผู้ใช้แต่ละประเภท (**Role-based Access Control**) เพื่อให้มั่นใจได้ว่าผู้ใช้ที่มีสิทธิ์เท่านั้นที่จะสามารถเข้าถึงข้อมูลสำคัญได้ นอกจากนี้ ผู้ดูแลระบบยังรับผิดชอบในการจัดการข้อมูลเกี่ยวกับการเรียนการสอน เช่น การจัดการภาคการศึกษา (**Semester Management**) การจัดการหลักสูตร (**Curriculum Management**) และการจัดการ **PLOs (Program Learning Outcomes)** ซึ่งจะช่วยให้ระบบสามารถสนับสนุนการเรียนการสอนได้อย่าง ผู้ดูแลระบบยังต้องดูแลการจัดการข้อมูลหลักสูตร (**Course Management**) เช่น การเพิ่ม/แก้ไขข้อมูลรายวิชา และการจัดการรายชื่ออาจารย์ผู้สอน

### 1.3.2 ขอบเขตด้านซอฟต์แวร์

ในโครงการนี้ เน้นการพัฒนาระบบที่สามารถใช้งานได้บนอุปกรณ์หลากหลายประเภท โดยมีการกำหนดขอบเขตการใช้งานของผู้ใช้ในแต่ละบทบาท ดังนี้:

- อาจารย์ผู้สอน (Instructor)** สามารถเข้าใช้งานการดู วิเคราะห์ข้อมูลคะแนน ค่าสถิติ ผ่านกราฟในแต่ละรายวิชา และการจัดทำเอกสาร **TQF3** และ **TQF5** โดยสามารถใช้งาน บนโทรศัพท์มือถือ แท็บเล็ต และคอมพิวเตอร์ ขึ้นอยู่กับลักษณะของฟังก์ชันที่ใช้งาน ในบางฟังก์ชันที่เกี่ยวข้องกับการแสดงกราฟต่างๆ เช่น กราฟการกระจายตัวคะแนน (Score Distribution Graph) และฮิสโตรีแกรม (Histogram) ซึ่งต้องการการแสดงผลที่มีพื้นที่เพียงพอในการแสดงรายละเอียดของข้อมูล จะต้องใช้งานผ่านสมาร์ทโฟน และแท็บเล็ตในแนวโน้ม (Landscape) การจัดทำเอกสาร **TQF3** และ **TQF5** จะต้องใช้งานผ่านแท็บเล็ตในแนวโน้ม (Landscape) และเดสก์ท็อป หรือแล็ปท็อป เท่านั้น ทั้งนี้ระบบรองรับการใช้งานในภาษาอังกฤษเท่านั้น ซึ่งหมายความว่าอาจารย์ผู้สอนชาวต่างชาติที่สอนในมหาวิทยาลัยเชียงใหม่ และจำกัดการใช้งานเฉพาะสำหรับบุคลากรของมหาวิทยาลัยเชียงใหม่เท่านั้น
- นักศึกษา (Student)** สามารถเข้าใช้งานการดู วิเคราะห์ข้อมูลคะแนน ค่าสถิติ ของตนเองผ่านกราฟในแต่ละรายวิชา โดยสามารถใช้งานผ่านสมาร์ทโฟน แท็บเล็ต และเดสก์ท็อป หรือแล็ปท็อป ขึ้นอยู่กับลักษณะของฟังก์ชันที่ใช้งาน ในบางฟังก์ชันที่เกี่ยวข้องกับการแสดงกราฟต่างๆ เช่น กราฟการกระจายตัวคะแนน (Score Distribution Graph), ฮิสโตรีแกรม (Histogram), และสไปเดอร์ชาร์ต (Spider Chart) ซึ่งต้องการการแสดงผลที่มีพื้นที่เพียงพอในการแสดงรายละเอียดของข้อมูล จะต้องใช้งานผ่านสมาร์ทโฟน และแท็บเล็ตในแนวโน้ม (Landscape) ทั้งนี้ระบบรองรับการใช้งานในภาษาอังกฤษเท่านั้น โดยการออกแบบนี้หมายความว่าบุคลากรของมหาวิทยาลัยเชียงใหม่ และจำกัดการใช้งานเฉพาะสำหรับบุคลากรของมหาวิทยาลัยเชียงใหม่เท่านั้น

3. นักจัดการงานทั่วไป (Department Staff) สามารถเข้าใช้งานและจัดการข้อมูลที่เกี่ยวข้องกับการจัดทำ และนำออกเอกสาร TQF และการดูการจับคู่ CLO (Course Learning Outcomes) และ PLO (Program Learning Outcomes) ของแต่ละรายวิชา โดยฟังก์ชันเหล่านี้รองรับการใช้งานผ่านแท็บเล็ตในแนวนอน (Landscape) และเดสก์ท็อป หรือแล็ปท็อป เท่านั้น เพื่อให้การแสดงผลข้อมูลที่ซับซ้อนและการวิเคราะห์ความสัมพันธ์ของ CLO และ PLO ในแต่ละรายวิชาเหมาะสมกับขนาดของหน้าจอและฟังก์ชันที่จำเป็น ทั้งนี้ระบบรองรับการใช้งานในภาษาอังกฤษเท่านั้น และจำกัดการใช้งานเฉพาะสำหรับบุคลากรของมหาวิทยาลัยเชียงใหม่เท่านั้น
4. กลุ่มผู้ดูแลระบบ (Admin) สามารถเข้าใช้งานการบริหารจัดการ และควบคุมระบบ โดยฟังก์ชันเหล่านี้รองรับการใช้งานผ่านแท็บเล็ตในแนวนอน (Landscape) และเดสก์ท็อป หรือแล็ปท็อป เพื่อให้การทำงานที่เกี่ยวข้องเหมาะสมกับขนาดของหน้าจอและฟังก์ชันที่จำเป็น ทั้งนี้ระบบรองรับการใช้งานในภาษาอังกฤษเท่านั้น และจำกัดการใช้งานเฉพาะสำหรับบุคลากรของมหาวิทยาลัยเชียงใหม่เท่านั้น

## 1.4 ประโยชน์ที่ได้รับ

1. ลดภาระการทำงานของอาจารย์และเจ้าหน้าที่ สามารถช่วยในการกรอกข้อมูลและจัดการเอกสารต่างๆ เช่น TQF3 และ TQF5 ทำให้อาจารย์และเจ้าหน้าที่ต้องใช้ในการทำงานเอกสารลงอย่างมาก
2. การวิเคราะห์ผลการเรียนรู้ที่แม่นยำ ระบบสามารถให้ข้อมูลที่เกี่ยวข้องกับผลการเรียนรู้ของนักศึกษาผ่านกราฟต่างๆ เช่น ฮิสโตแกรม (Histogram) และกราฟการกระจายตัวคะแนน (Score Distribution Graph) ซึ่งช่วยให้อาจารย์สามารถวิเคราะห์ข้อมูลได้ง่ายและแม่นยำยิ่งขึ้น
3. เพิ่มประสิทธิภาพในการจัดการข้อมูลการศึกษาของนักศึกษา สามารถจัดการข้อมูลคณิตและผลการเรียนรู้ของนักศึกษามีความเป็นระเบียบและสามารถเข้าถึงได้ง่ายขึ้น โดยมีการเชื่อมโยงระหว่างคะแนนนักศึกษากับ CLO (Course Learning Outcomes) และ PLO (Program Learning Outcomes) ที่ชัดเจน
4. สนับสนุนการตัดสินใจในการปรับปรุงการเรียนการสอน สามารถเข้าถึงและสามารถวิเคราะห์ผลการเรียนรู้ได้ง่าย อาจารย์และเจ้าหน้าที่สามารถใช้ข้อมูลนี้ในการปรับปรุงแผนการเรียนการสอน โดยการปรับระดับความยากง่ายของข้อสอบหรือการปรับวิธีการสอนเพื่อให้เหมาะสมกับผลการเรียนรู้ของนักศึกษา
5. เพิ่มความสะดวกในการเข้าถึงข้อมูลสำหรับนักศึกษา สามารถเข้าถึงผลคณิตและข้อมูลที่เกี่ยวข้องกับการเรียนได้ง่ายขึ้น ไม่ว่าจะเป็นการตรวจสอบคณิตจากการสอบหรือการดูการพัฒนาผลการเรียนรู้ของตนเองตลอดระยะเวลาการเรียน ผ่านสไปเดอร์ชาร์ต (Spider Chart) ซึ่งจะช่วยให้นักศึกษาสามารถติดตามความก้าวหน้าในการเรียนรู้ของตนได้อย่างชัดเจน

## 1.5 เทคโนโลยีและเครื่องมือที่ใช้

### 1.5.1 เทคโนโลยีด้านฮาร์ดแวร์

1. คอมพิวเตอร์เซิร์ฟเวอร์ (Servers) เซิร์ฟเวอร์ที่มีความสามารถในการรองรับการประมวลผลข้อมูล พร้อมทั้งการจัดเก็บฐานข้อมูล เซิร์ฟเวอร์เหล่านี้จะถูกตั้งอยู่ในศูนย์ข้อมูลของมหาวิทยาลัยหรือคลา

วัดที่มีการเชื่อมต่อเครือข่าย

- คอมพิวเตอร์สำหรับการพัฒนาและออกแบบ (**Development and Design Computers**) คอมพิวเตอร์ แล็ปท็อป เช่น เครื่องที่มีโปรเซสเซอร์หลายคอร์, หน่วยความจำ (RAM), และพื้นที่เก็บข้อมูลที่ (SSD) เพื่อให้สามารถพัฒนาและทดสอบระบบ
- แท็บเล็ตสำหรับการวาดรูปและออกแบบโลโก้ (**Tablets for Drawing and Logo Design**) แท็บเล็ตจะถูกใช้ในการวาดรูปและออกแบบโลโก้ของเว็บแอปพลิเคชัน ScoreOBE+ ซึ่งช่วยให้การออกแบบภาพประกอบและโลโก้ที่มีความละเอียด
- อุปกรณ์สำหรับการทดสอบ (**Testing Devices**) ในการทดสอบการทำงานของระบบ ScoreOBE+ จะใช้โทรศัพท์มือถือ, แท็บเล็ต, และคอมพิวเตอร์ที่มีขนาดจอกลมและเบราว์เซอร์ต่างกัน เพื่อทดสอบการทำงานของระบบในสภาพแวดล้อมที่หลากหลาย โดยอุปกรณ์เหล่านี้จะต้องรองรับการทำงานในหลายระบบปฏิบัติการ เช่น iOS, Android, Windows และ MacOS รวมทั้งเบราว์เซอร์ที่หลากหลาย เช่น Safari, Chrome, และ Edge เพื่อให้มั่นใจว่าแอปพลิเคชันสามารถทำงานได้บนแพลตฟอร์มต่างๆ

### 1.5.2 เทคโนโลยีด้านซอฟต์แวร์

- Figma**[5]: เครื่องมือออกแบบที่ใช้สำหรับการสร้างดีไซน์ สร้างต้นแบบ (Prototype) และจัดการระบบการออกแบบ (Design System)
- Procreate**[26]: แอปพลิเคชันวาดภาพและออกแบบกราฟิกที่ได้รับความนิยมบน iPad
- TypeScript**[16]: ภาษาการเขียนโปรแกรมที่เป็นส่วนขยายของ JavaScript เพิ่มความสามารถในการตรวจสอบประเภทข้อมูล (type) เพื่อช่วยลดข้อผิดพลาดและเพิ่มความสามารถในการอ่านและบำรุงรักษาโค้ด
- React**[20]: ไลบรารีสำหรับสร้างส่วนติดต่อผู้ใช้ (UI) โดยเน้นการสร้างคอมโพเนนต์ (Component) ที่สามารถกลับมาใช้ซ้ำได้ แต่ละคอมโพเนนต์สามารถจัดการสถานะ (state) และเมธอด (methods) เพื่อการจัดการข้อมูลและการแสดงผล
- Tailwind CSS**[16]: เฟรมเวิร์ก CSS ที่ให้คลาสที่กำหนดไว้ล่วงหน้า ช่วยให้นักพัฒนาสามารถสร้างดีไซน์ได้อย่างรวดเร็วและสม่ำเสมอ
- Mantine**[13]: ไลบรารีคอมโพเนนต์ React ที่มีสไต์ล์และฟังก์ชันการทำงานที่ครบครัน ช่วยให้นักพัฒนาสามารถสร้าง UI ได้อย่างรวดเร็ว
- Redux**[24]: ไลบรารีสำหรับจัดการสถานะ (state) ของแอปพลิเคชัน JavaScript โดยเฉพาะกับ React ช่วยให้การจัดการข้อมูลในแอปพลิเคชันมีความเป็นระเบียบและคาดการณ์ได้
- Chart.js**[2]: ไลบรารี JavaScript ที่ใช้สร้างกราฟและแผนภูมิแบบตอบสนอง ช่วยในการแสดงข้อมูลในรูปแบบภาพ
- Framer Motion**[4][10]: ไลบรารีสำหรับสร้างแอนิเมชันใน React ช่วยให้นักพัฒนาสามารถเพิ่มการเคลื่อนไหวที่ราบรื่นและซับซ้อนได้อย่างง่ายดาย

10. **NestJS[19]**: เฟรมเวิร์กสำหรับการสร้างแอปพลิเคชันฝั่งเซิร์ฟเวอร์โดยใช้ Node.js ออกแบบมาเพื่อเพิ่มประสิทธิภาพและความสามารถในการบำรุงรักษา
11. **Passport.js[22]**: มิเดเดิลแวร์สำหรับ Node.js ที่ใช้ในการตรวจสอบและจัดการการยืนยันตัวตน (authentication) ของผู้ใช้
12. **JSON Web Token (JWT)[22]**: มาตรฐานเปิดสำหรับการแลกเปลี่ยนข้อมูลที่ปลอดภัยระหว่างฝ่ายต่าง ๆ ในรูปแบบของ JSON ใช้ในการยืนยันตัวตนและการอนุญาต
13. **Swagger[28]**: เครื่องมือที่ช่วยในการออกแบบ สร้างเอกสาร และทดสอบ API ทำให้การพัฒนาและบำรุงรักษา API เป็นไปอย่างมีประสิทธิภาพ
14. **PDFKit[23]**: ไลบรารี JavaScript ที่ใช้ในการสร้างและจัดการไฟล์ PDF บนฝั่งเซิร์ฟเวอร์
15. **Mongoose[15]**: ไลบรารี Object Data Modeling (ODM) สำหรับ MongoDB และ Node.js ช่วยในการจัดการและตรวจสอบข้อมูลในฐานข้อมูล
16. **MongoDB[14]**: ระบบฐานข้อมูล NoSQL ที่จัดเก็บและดึงข้อมูลด้วยวิธีที่ยืดหยุ่นและปรับขนาดได้
17. **Postman[25]**: เครื่องมือที่ใช้ทดสอบการทำงานของ API ช่วยให้นักพัฒนาสามารถส่งคำขอและดูการตอบสนองได้อย่างง่ายดาย
18. **Docker[3]**: แพลตฟอร์มโอเพนซอร์สที่ช่วยในการสร้าง ทดสอบ และปรับใช้แอปพลิเคชันในรูปแบบของคอนเทนเนอร์ ซึ่งเป็นสภาพแวดล้อมที่แยกจากกันที่สามารถรันแอปพลิเคชันได้

## 1.6 แผนการดำเนินงาน

ขั้น ตอน การ ดำเนินงาน	พ.ย. 2566	ธ.ค. 2566	ม.ค. 2567	ก.พ. 2567	มี.ย. 2567	พ.ค. 2567	มิ.ย. 2567	ก.ค. 2567	ส.ค. 2567	ก.ย. 2567	ต.ค. 2567	พ.ย. 2567	ธ.ค. 2568	ม.ค. 2568	ก.พ. 2568	มี.ย. 2568	
เลือก หัวข้อ โครงการ																	
เลือกอาจารย์ ที่ปรึกษา																	
สำรวจ ความ ต้องการ ของ ผู้ใช้																	
ศึกษา ระบบ CPE QA																	

ชั้น ตอน การ ดำเนินงาน	พ.ย. 2566	ธ.ค. 2566	ม.ค. 2567	ก.พ. 2567	มี.ค. 2567	เม.ย. 2567	พ.ค. 2567	มิ.ย. 2567	ก.ค. 2567	ส.ค. 2567	ก.ย. 2567	ต.ค. 2567	พ.ย. 2567	ธ.ค. 2568	ม.ค. 2568	ก.พ. 2568	มี.ค. 2568
ออกแบบ UX/ UI และ feature ต่างๆ																	
พัฒนา Backend																	
ออกแบบฐาน ข้อมูล																	
พัฒนา Frontend																	
ทดสอบ กับ ผู้ ใช้งานจริง																	

## 1.7 บทบาทและความรับผิดชอบ

- นางสาว รนพร ชาญชนะโยธิน (**Thanaporn Chanchanayotin**) ทำหน้าที่เป็น Full-stack Developer และออกแบบฐานข้อมูล รวมถึงพัฒนาระบบหลังบ้าน (Backend Development) โดยรับผิดชอบการออกแบบโครงสร้างฐานข้อมูล การเชื่อมต่อกับ API และการจัดการการเก็บข้อมูล การพัฒนาและดูแลส่วนของ server-side ให้ระบบสามารถทำงานได้
- นางสาว วรพิชชา เมืองยศ (**Worapitcha Muengyot**) ทำหน้าที่ในการออกแบบและพัฒนาส่วนติดต่อผู้ใช้ (Frontend Development) โดยรับผิดชอบออกแบบหน้าเว็บแอปพลิเคชัน ประสบการณ์ผู้ใช้ และการพัฒนาฟังก์ชันต่างๆ
- นาย สวิช จาเริกพุนผล (**Switch Jarukpunpon**) ทำหน้าที่ในการออกแบบและพัฒนาส่วนติดต่อผู้ใช้ (Frontend Development) โดยรับผิดชอบในการออกแบบหน้าเว็บ และประสบการณ์ผู้ใช้ และพัฒนาฟังก์ชันต่างๆ และรวมถึงการทดสอบการใช้งานฟังก์ชันต่างบนเว็บ

## 1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

โครงการ ScoreOBE+ มีผลกระทบหลายด้านที่สามารถช่วยปรับปรุงการจัดการและการวิเคราะห์ข้อมูล ทางการศึกษาในมหาวิทยาลัย โดยมีผลกระทบต่อด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม ดังนี้:

การใช้ระบบ ScoreOBE+ จะช่วยให้อาจารย์และนักศึกษามีเครื่องมือที่สะดวกในการติดตามผลการเรียนและการประเมินผลได้อย่างมีประสิทธิภาพ ซึ่งสามารถสร้างความโปร่งใสในการให้คะแนนและการติด-

ตามพัฒนาการของนักศึกษา ส่งผลให้มีการกระจายข้อมูลที่เท่าเทียมกันระหว่างอาจารย์และนักศึกษา นอก-จากนี้ยังสามารถสนับสนุนการพัฒนาทักษะของนักศึกษาผ่านการนำเสนอข้อมูลการเรียนรู้ที่มีประสิทธิภาพ และสะดวกในการเข้าถึง

การใช้ระบบนี้ไม่เพียงแค่ส่งผลดีต่อการจัดการข้อมูลทางการศึกษา แต่ยังช่วยลดภาระการทำงานที่มาก เกินไปจากการจัดการคะแนนด้วยตนเอง ซึ่งอาจส่งผลดีต่อสุขภาพจิตของอาจารย์และบุคลากรที่เกี่ยวข้อง การลดภาระงานที่ซ้ำซากและเพิ่มประสิทธิภาพในการทำงานสามารถช่วยลดความเครียดที่เกิดจากการที่มีความซับซ้อน

ระบบ ScoreOBE+ จะถูกออกแบบให้มีมาตรการความปลอดภัยที่เหมาะสมในการป้องกันข้อมูลส่วนบุคคลของนักศึกษาและอาจารย์ โดยข้อมูลการเรียนและการเรียนจะถูกเก็บรักษาในเซิร์ฟเวอร์ของภาควิชาศิวกรรมคอมพิวเตอร์ คณะศิวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ซึ่งช่วยป้องกันการเข้าถึงที่ไม่ได้รับอนุญาต

เนื่องจากระบบนี้เกี่ยวข้องกับการรวบรวมข้อมูล การใช้ข้อมูล การเปิดเผยข้อมูล และการรักษาข้อมูลส่วนบุคคลของนักศึกษาและอาจารย์ของมหาวิทยาลัยเชียงใหม่ จึงต้องปฏิบัติตามตามนโยบายคุ้มครองข้อมูลส่วนบุคคล มหาวิทยาลัยเชียงใหม่ (CMU Privacy Policy) ทั้งนี้ระบบมีการขออนุญาตและการจัดเก็บข้อมูลอย่างเหมาะสมตามข้อกำหนดทางกฎหมาย

การใช้ระบบนี้ช่วยสร้างความโปร่งใสในการให้คะแนนและการติดตามผลการเรียนรู้ ซึ่งอาจส่งผลดีต่อการเปลี่ยนแปลงทางวัฒนธรรมการเรียนรู้ในมหาวิทยาลัย โดยช่วยส่งเสริมการเรียนรู้ที่มีคุณภาพและสามารถปรับตัวตามความต้องการของนักศึกษาและคณาจารย์ได้อย่างเหมาะสม ระบบจะสนับสนุนการเข้าถึงข้อมูลและการเรียนรู้ในรูปแบบที่ทันสมัย และช่วยให้การเรียนการสอนมีความยืดหยุ่นและเป็นมิตรกับผู้เรียนมากขึ้น

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 ความหมายของ OBE

การศึกษาที่เน้นผลลัพธ์ (Outcome-based Education - OBE) เป็นแนวทางการศึกษาที่มุ่งเน้นการกำหนดเป้าหมายหรือผลลัพธ์การเรียนรู้สำหรับนักเรียน โดยมุ่งเน้นที่จะช่วยพัฒนาให้ผู้เรียนมีความรู้ ทักษะ และบุคลิกภาพต่าง ๆ อันจะทำให้สัมฤทธิผลหรือบรรลุถึงผลลัพธ์ตามที่หลักสูตรได้กำหนดไว้ โดย OBE จะมีการวัดผลลัพธ์และการพัฒนาอย่างต่อเนื่อง เป็นการเก็บรวบรวมข้อมูลของผลลัพธ์การเรียนรู้ต่าง ๆ ที่ผู้เรียนได้สัมฤทธิผล แล้วใช้ข้อมูลวัดผลลัพธ์นั้นไปพัฒนาและเพิ่มเติมหลักสูตร การหรือกระบวนการวิชาต่อไป [21]

#### 2.2 แพลตฟอร์มที่คล้ายคลึงกัน

##### 2.2.1 CPE QA

CPE QA เป็นแพลตฟอร์มสำหรับการจัดทำ มาตรฐานคุณภาพ (TQF3) และ มาตรฐานคุณภาพ (TQF5) ในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์มหาวิทยาลัยเชียงใหม่ โดยจะมีผู้ใช้อยู่ 2 บทบาท คือผู้สอน และผู้ดูแลระบบ ผู้สอนมีหน้าที่กรอก มาตรฐาน 3 เมื่อจะทำการเปิดกระบวนการวิชานั้นๆ เพื่อกำหนดแนวทางการสอนและการเก็บคะแนนในกระบวนการวิชาที่ตนเองจะสอน และเมื่อสิ้นสุดเทอมการศึกษาต้องจัดทำ มาตรฐาน 5 เพื่อประเมินผลการเรียนรู้ของผู้เรียนในกระบวนการวิชาที่ตนเองเปิดสอน โดยระบบจะแบ่งการกรอก มาตรฐาน 3 เป็น 5 ส่วน คือ

1. หมวดข้อมูลทั่วไป (General Information) ผู้สอนต้องกรอกประเพณีกระบวนการวิชา ลักษณะกระบวนการวิชา รูปแบบการประเมินผลของกระบวนการวิชา ภาคการศึกษาที่สอน ชั้นปีที่เรียน ชื่ออาจารย์ ผู้สอนทั้งหมด ชื่อผู้ประสานงานกระบวนการวิชา สถานที่สอนค่าบบรายวัน หรือสถานที่เรียนค่าตอบแทน
2. หมวดทรัพยากรประกอบการเรียนการสอน (Course Materials) ผู้สอนต้องกรอกตำราและเอกสารหลัก และเอกสารแนะนำ
3. หมวดคำอธิบายลักษณะกระบวนการวิชาและแผนการสอน (Description and Planning) ผู้สอนจะต้องกรอกผลลัพธ์การเรียนรู้ระดับกระบวนการวิชาทั้งภาษาไทยและภาษาอังกฤษ (Course Learning Outcomes, CLOs) และเนื้อหาวิชาและแผนการสอน โดยจะต้องกรอกชื่อหัวเรื่องและจำนวนชั่วโมงที่จะสอนในอาทิตย์นั้น
4. หมวดการเชื่อมโยงผลลัพธ์การเรียนรู้ระดับกระบวนการวิชา กับผลลัพธ์การเรียนรู้ระดับหลักสูตร (Curriculum Mapping) ผู้สอนต้องเชื่อมโยงผลลัพธ์การเรียนรู้ระดับกระบวนการวิชาของกระบวนการวิชาตนเอง เข้ากับผลลัพธ์การเรียนรู้ระดับหลักสูตร (Program Learning Outcomes, PLOs) ผลลัพธ์การเรียนรู้ระดับกระบวนการวิชาจะต้องเชื่อมโยงกับผลลัพธ์การเรียนรู้ระดับหลักสูตรอย่างน้อย 1 หัวข้อ และผลลัพธ์การเรียนรู้ระดับหลักสูตรอ้างอิงตามเกณฑ์ของ Accreditation Board for Engineering and Technology (ABET) เป็นเกณฑ์มาตรฐานเพื่อวัดคุณภาพการศึกษาวิศวกรรมศาสตร์บัณฑิตของสถาบันอุดมศึกษาไทย ซึ่งสามารถนำไปใช้ศึกษาและพัฒนาการดำเนินงานของหลักสูตรให้มีความสอดคล้องและเป็นไปตามนโยบายขั้นตอนและเกณฑ์มาตรฐานที่ ABET กำหนดไว้อย่างมีประสิทธิภาพ

5. หมวดการประเมินผลของกระบวนการวิชา (Course Assessment) ผู้สอนต้องกรอกเครื่องมือการประเมิน (Assessment Tool) ในกระบวนการวิชาที่ตนสอน เช่น การสอบกลางภาค หรือการสอบปลายภาค ผู้สอนต้องเชื่อมโยงเครื่องมือการประเมินเข้ากับผลลัพธ์การเรียนรู้ระดับวิชา และเครื่องมือการประเมินคิดแต่ละเครื่องมือคิดเป็นกีเปอร์เซ็นต์ของการประเมินในกระบวนการวิชานั้นๆ

มคอ.5 เป็น 3 ส่วน คือ

1. หมวดการประเมินกระบวนการวิชา (Course Evaluation) ผู้สอนต้องกรอกเลขเช็คชั่นในกระบวนการวิชาที่ตนสอน และต้องกรอก จำนวนนักศึกษาที่ได้เกรด A ถึง F ในแต่ละเช็คชั่น ซึ่งระบบ CPE QA จะคำนวณเกรดเฉลี่ยในแต่ละเช็คชั่นให้อัตโนมัติ นอกจากนี้ผู้สอนต้องระบุเกณฑ์การตัดเกรด (Grading Criteria) และช่วงคะแนนในการตัดเกรด .
2. หมวดเกณฑ์การประเมินตามวัตถุประสงค์เฉพาะของกระบวนการกระบวนการวิชา (Rubrics for CSO evaluation) ผู้สอนต้องกรอกจำนวนนักศึกษาในแต่ละผลลัพธ์การเรียนรู้ระดับกระบวนการวิชาที่ได้คะแนนตั้งแต่ 0 ถึง 4 โดยระบุว่ามีนักศึกษาเกี่ยวกันที่ได้คะแนนในแต่ละระดับ
3. หมวดการประเมินตามวัตถุประสงค์เฉพาะของกระบวนการกระบวนการวิชา (CSO Evaluation) ผู้สอนต้องกรอกเครื่องมือการประเมินในกระบวนการวิชาที่ตนสอนเพื่อเชื่อมโยงกับผลลัพธ์การเรียนรู้ระดับกระบวนการวิชา และจะต้องระบุเลขข้อในเครื่องมือการประเมิน ซึ่งระบบ CPE QA จะคำนวณคะแนนเฉลี่ยผลลัพธ์การเรียนรู้ระดับกระบวนการวิชาจาก มคอ.5 ส่วน 2

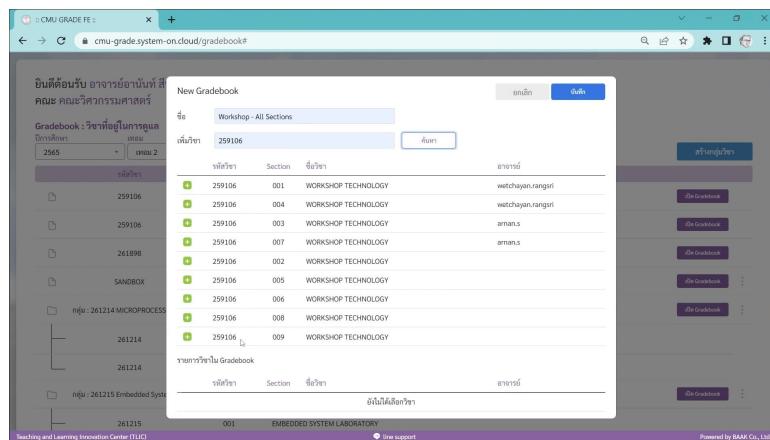
เมื่อผู้สอนกรอกเสร็จ ผู้ดูแลระบบจะพิมพ์ทั้งสองเอกสารเพื่อทำและนำส่งให้กับทางคณบดี วิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

### 2.2.2 CMU Score Announcement

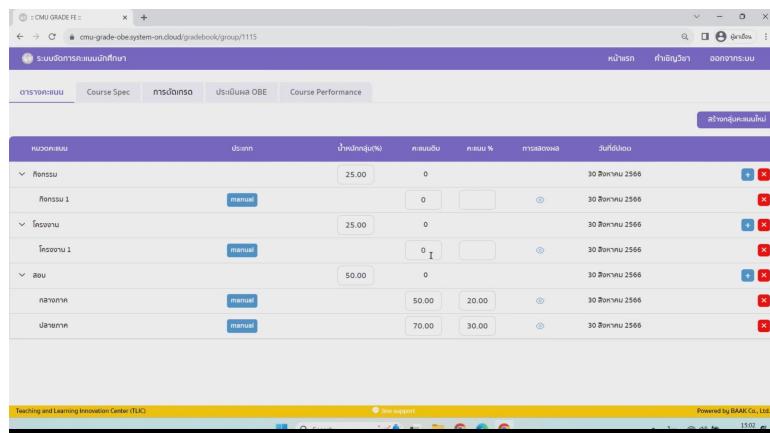
CMU Score Announcement เป็นแพลตฟอร์มสำหรับประกาศคะแนนของมหาวิทยาลัยเชียงใหม่ ผู้สอนต้องเพิ่มกระบวนการวิชาและเลขเช็คชั่นที่ต้องการประกาศ หลังจากเพิ่มกระบวนการวิชาเสร็จแล้ว ผู้สอนต้องนำเข้าคะแนนในรูปแบบสเปรตชีต ระบบมีแม่แบบการนำเข้าคะแนน หลังจากนำเข้าคะแนนเสร็จ ระบบจะคำนวณค่าทางสถิติให้โดยอัตโนมัติ ถ้าผู้สอนกรอกคะแนนผิด สามารถแก้ไขคะแนนรายบุคคลในระบบโดยไม่ต้องแก้ไขในสเปรตชีต ระบบก็จะคำนวณค่าทางสถิติใหม่ สำหรับการประกาศคะแนน ผู้สอนสามารถเลือกประกาศทุกเช็คชั่นหรือในบางเช็คชั่น นักศึกษาจะเห็นคะแนนของตนเองพร้อมค่าทางสถิติและการฟอกคะแนนของงานมอบหมายนั้นๆ

### 2.2.3 CMU OBE

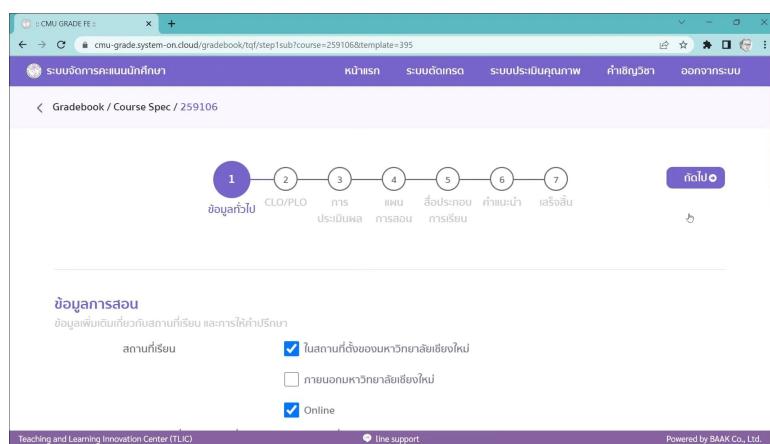
CMU OBE เป็นแพลตฟอร์มสำหรับช่วยในการประเมินผลลัพธ์การเรียนรู้ (OBE) ผู้สอนสามารถสร้าง Gradebook โดยเลือกจากวิชาที่ตนเป็นผู้สอน และหากวิชาที่ต้องการประเมินผล OBE มีหลายเช็คชั่นหรือมีผู้สอนหลายคน และต้องการที่จะประเมินผล OBE ร่วมกัน สามารถสร้างกลุ่มของวิชานั้น และเชิญผู้สอนท่านอื่นได้ และสุดท้ายผู้สอนสามารถสร้าง มคอ.3 (TQF3) ของวิชานั้นๆ ได้



รูปที่ 2.1: Create Gradebook



รูปที่ 2.2: Create Score

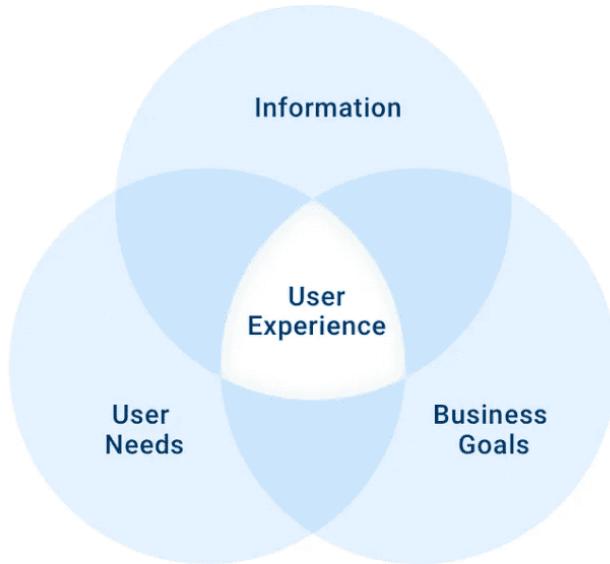


รูปที่ 2.3: Create TQF3

## 2.3 User Experience and User Interface

### 2.3.1 User-Centered Design (UCD)

เป็นหลักการการออกแบบที่จัดระดับความสำคัญของความต้องการและความชอบของผู้ใช้ตลอดกระบวนการออกแบบ โดยการทำความเข้าใจผู้ใช้เป้าหมายและบริบทของผู้ใช้ เพื่อสร้างเว็บแอปพลิเคชันที่ใช้งานง่ายมีประสิทธิภาพในการใช้งาน โดยทั่วไปหลักการนี้จะเกี่ยวข้องกับการออกแบบช้าๆ เช่น การวิจัยผู้ใช้ การสร้างต้นแบบ และการทดสอบการใช้งานเพื่อปรับปรุงประสบการณ์ของผู้ใช้อย่างต่อเนื่อง [29] [30]



รูปที่ 2.4: How to design in User-Centered Way

### 2.3.2 Jakob's Law

เป็นหลักการการออกแบบว่าด้วยการพัฒนาหรือปรับปรุงผลิตภัณฑ์ไม่ว่าจะเป็นสิ่งของ หรือเว็บไซต์ แอปพลิเคชันจากความคุ้นเคยเดิมของผู้ใช้ ให้สามารถใช้งานผลิตภัณฑ์ได้ง่ายและสะดวกมากขึ้นโดยที่ไม่ต้องเริ่มเรียนรู้การใช้งานใหม่อีกครั้ง โดยจะใช้รูปแบบ UI ที่ผู้ใช้คุ้นเคย เพื่อไม่ให้ผู้ใช้ต้องปรับตัวใหม่ หรือการจัดองค์ประกอบของส่วนต่างๆ ในหน้า layout ให้เหมาะสม ไม่พยายามเปลี่ยนแปลงประสบการณ์ของผู้ใช้มากจนเกินไป ซึ่งความสอดคล้องตรงนี้จะช่วยให้ผู้ใช้รู้สึกสบายและมั่นใจในการใช้ผลิตภัณฑ์มากขึ้น เมื่องค์ประกอบบนเว็บไซต์ต้องทำงานสอดคล้องกับความคาดหวังของผู้ใช้งาน [11] [12]

### 2.3.3 Gestalt Principles

ทฤษฎีเกสตัลท์ (Gestalt Principles) เป็นแนวคิดทางจิตวิทยาที่เกี่ยวข้องกับการรับรู้ภาพ (Visual Perception) โดยสมองของมนุษย์มักจะมองเห็นภาพรวมทั้งหมดก่อนจะมองรายละเอียดเป็นๆ กัน หลักการนี้ถูกนำมาใช้ในการออกแบบส่วนติดต่อผู้ใช้ (UI) และประสบการณ์ผู้ใช้ (UX) เพื่อช่วยให้การนำเสนอข้อมูลมีประสิทธิภาพยิ่งขึ้น ซึ่งในระบบ ScoreOBE+ เรายังนำหลักการเกสตัลท์มาใช้ในการออกแบบให้ผู้ใช้สามารถเข้าใจข้อมูลคงแนะนำและผลลัพธ์การเรียนรู้ได้ง่ายขึ้น หลักการเกสตัลท์ที่สำคัญมีทั้งหมด 7 ข้อ ดังนี้ [6] [7]

1. **การปิด (Closure)** หลักการนี้ระบุว่าสมองของมนุษย์มักจะเติมเต็มช่องว่างของรูปร่างหรือองค์ประกอบที่ไม่สมบูรณ์ เพื่อให้มองเห็นเป็นภาพที่สมบูรณ์ เช่น การเห็นวงกลมที่ขาดบางส่วนเป็นวงกลมเต็ม ตัวอย่างใน ScoreOBE+: - การออกแบบไอคอนหรือกราฟที่ใช้เส้นประหรือเส้นขาดช่วง เพื่อให้ผู้ใช้สามารถคาดเดารูปทรงหรือแนวโน้มของข้อมูลที่เหลืออยู่
2. **การคล้ายคลึง (Similarity)** หลักการนี้ระบุว่าสมองของมนุษย์จะจัดกลุ่มองค์ประกอบที่มีลักษณะคล้ายกันเข้าด้วยกัน โดยพิจารณาจากสี รูปร่าง หรือขนาด ตัวอย่างใน ScoreOBE+: - ใช้สีเดียวกัน สำหรับหมวดหมู่ของคะแนน เช่น คะแนนของตัวเองเป็นสีน้ำเงิน คะแนนค่าทางสถิติจะเป็นสีดำ เพื่อให้เข้าใจง่ายขึ้น
3. **ความใกล้ชิด (Proximity)** หลักการนี้ระบุว่าองค์ประกอบที่อยู่ใกล้กันมักจะถูกมองว่าเป็นกลุ่มเดียวกัน ตัวอย่างใน ScoreOBE+: - การจัดกลุ่มคะแนนของแต่ละหมวด เช่น คะแนนสอบ คะแนนแบบฝึกหัด และคะแนนโครงการ ไว้ใกล้กัน เพื่อลดความสับสนของผู้ใช้
4. **ความต่อเนื่อง (Continuity)** สมองของมนุษย์มักจะมองหาเส้นทางหรือรูปแบบที่ต่อเนื่องกัน แม้ว่าจะมีองค์ประกอบอื่นรบกวน ตัวอย่างใน ScoreOBE+: - การจัดวางปุ่มที่เกี่ยวข้องกันให้อยู่ใกล้กัน เช่น ปุ่ม ”ยกเลิก” และ ”บันทึกคะแนน” อยู่ติดกันเพื่อลดความผิดพลาดในการใช้งาน
5. **พิกเกอร์-กราวด์ (Figure-Ground)** หลักการนี้เกี่ยวข้องกับการแยกแยะระหว่างวัตถุ (Figure) และพื้นหลัง (Ground) ซึ่งช่วยให้สามารถโฟกัสที่ข้อมูลสำคัญได้ ตัวอย่างใน ScoreOBE+: - การใช้เงาล้อมรอบการ์ตดของกระบวนการวิชา เพื่อให้วิชาที่แสดงเด่นชัดมากขึ้น
6. **สมมาตร (Symmetry)** สมองของมนุษย์ชอบรูปแบบที่สมมาตร เพราะช่วยให้ข้อมูลดูเป็นระเบียบ และเข้าใจง่าย ตัวอย่างใน ScoreOBE+: - การออกแบบหน้าต่างแสดงคะแนนที่มีคอลัมน์และแถว สมดุลกัน ทำให้ข้อมูลดูเป็นระเบียบ
7. **ชะตากรรมร่วมกัน (Common Fate)** องค์ประกอบที่เคลื่อนที่ไปในทิศทางเดียวกันจะถูกมองว่าเป็นกลุ่มเดียวกัน ตัวอย่างใน ScoreOBE+: - การเลื่อน (scroll) รายการคะแนนของนักศึกษาแบบแนวตั้ง โดยให้แบบหัวข้อคงที่ด้านบน เพื่อให้ผู้ใช้สามารถติดตามชื่อคอลัมน์ได้ตลอดเวลา

การนำหลักการเกสตัลท์มาใช้ในการออกแบบ ScoreOBE+ ทำให้ระบบสามารถนำเสนอข้อมูลได้อย่างมีประสิทธิภาพ ช่วยให้ผู้ใช้เข้าใจข้อมูลได้เร็วขึ้นและใช้งานได้สะดวกขึ้น

#### 2.3.4 Interaction Design Principle

เป็นหลักการที่ใช้ในการออกแบบการจัดการการแสดงผลและปฏิสัมพันธ์ระหว่างผู้ใช้กับผลิตภัณฑ์ ซึ่งมุ่งเน้นให้ผู้ใช้มีประสบการณ์การใช้งานที่เป็นมิตรและเข้าใจได้ง่าย [8] [9]

1. **ทัศนวิสัย (Visibility)** ต้องการให้สิ่งที่สำคัญและที่ผู้ใช้ต้องการจะทราบเห็นได้่ายเพื่อให้เกิดความเข้าใจที่ถูกต้องและลดความสับสนในขณะการใช้งาน
2. **ข้อเสนอแนะ (Feedback)** การให้ผู้ใช้ทราบถึงผลการทำงานของตน เช่น การแสดงข้อความหรือการเปลี่ยนแปลงสถานะของระบบ

3. **ข้อจำกัด (Constraint)** การใช้ข้อจำกัดเพื่อป้องกันการกระทำที่ผู้ใช้ทำผิดพลาดต่อระบบ อาจเป็นข้อจำกัดในการใช้งานหรือการแสดงผล
4. **เข้าถึงได้ (Affordance)** การออกแบบให้เห็นว่าต้องสามารถใช้งานได้อย่างไร เช่น ปุ่มที่ออกแบบดูเหมือนจะกดได้

### 2.3.5 Design Thinking

Design Thinking เป็นแนวทางในการออกแบบที่มุ่งเน้นไปที่ผู้ใช้ (User-Centered Design) โดยมีเป้าหมายเพื่อสร้างนวัตกรรมที่สามารถแก้ไขปัญหาได้อย่างมีประสิทธิภาพ ผ่านกระบวนการ 5 ขั้นตอน ได้แก่ การทำความเข้าใจปัญหา (Empathize), การกำหนดปัญหา (Define), การระดมความคิด (Ideate), การสร้างต้นแบบ (Prototype) และการทดสอบ (Test) ซึ่งสามารถนำมาประยุกต์ใช้กับการพัฒนา **Score-OBE+** ได้ดังนี้

- **Empathize (ทำความเข้าใจปัญหา)**

ในขั้นตอนนี้ ทีมพัฒนา ScoreOBE+ ได้ทำการสำรวจปัญหาและความต้องการของผู้ใช้หลัก ได้แก่ อาจารย์ นักศึกษา และเจ้าหน้าที่ฝ่ายวิชาการ ผ่านการสัมภาษณ์และการเก็บข้อมูลจากระบบเดิม เช่น ระบบ CPE QA เพื่อเข้าใจปัญหาที่เกิดขึ้น เช่น การประภาคคคะแนนที่ยุ่งยาก หรือการเชื่อมโยงผลการเรียนรู้ (Learning Outcomes) ที่ไม่ชัดเจน

- **Define (กำหนดปัญหา)**

จากการเก็บข้อมูล พบว่าอาจารย์มีปัญหาในการอัปโหลดและวิเคราะห์คะแนนเพื่อประเมินผลสัมฤทธิ์ของผู้เรียน นักศึกษามีความต้องการในการเข้าถึงผลการเรียนได้สะดวกและเข้าใจข้อมูลสถิติของตนเองได้ง่ายขึ้น จึงได้กำหนดปัญหาหลักของ ScoreOBE+ ว่า “ระบบปัจจุบันไม่สามารถช่วยให้ผู้ใช้มองเห็นภาพรวมของผลการเรียนรู้ได้อย่างชัดเจนและมีประสิทธิภาพ”

- **Ideate (ระดมความคิด)**

ทีมพัฒนาได้ระดมความคิดเกี่ยวกับแนวทางแก้ไข โดยนำเสนอพังก์ชันใหม่ เช่น การสร้างกราฟสถิติ คณิตศาสตร์ (Histogram, Spider Chart), การแสดงผล CLO-PLO Mapping, และการแจ้งเตือนให้นักศึกษาทราบผลคะแนนแบบเรียลไทม์ ทั้งนี้ได้ทำการเปรียบเทียบแนวทางที่เป็นไปได้ เพื่อเลือกโซลูชันที่เหมาะสมที่สุด

- **Prototype (สร้างต้นแบบ)**

หลังจากกำหนดแนวทางแก้ไข ทีมงานได้ออกแบบต้นแบบของ ScoreOBE+ โดยเริ่มจาก Wireframe และ UI Mockups เพื่อให้เห็นภาพรวมของหน้าตาระบบ จากนั้นพัฒนา MVP (Minimum Viable Product) ที่ประกอบด้วยพังก์ชันพื้นฐาน เช่น การอัปโหลดคะแนน และการแสดงผลสถิติในเบื้องต้น เพื่อให้ผู้ใช้สามารถทดสอบได้

- **Test (ทดสอบ)**

ระบบต้นแบบถูกนำไปทดสอบกับอาจารย์และนักศึกษาจำนวน 810 คน เพื่อเก็บฟีดแบค โดยพบว่าผู้ใช้ต้องการให้มีการแสดงผลข้อมูลคะแนนที่ชัดเจนขึ้น และต้องการให้ระบบรองรับหลายแพลตฟอร์ม

ได้ดีขึ้น เช่น การถ่ายภาพในโหมด Landscape บนอุปกรณ์พกพา ที่มีพัฒนาได้มาข้อเสนอแนะมาปรับปรุงระบบก่อนเข้าสู่การพัฒนาเวอร์ชันสมบูรณ์

Design Thinking ช่วยให้ ScoreOBE+ สามารถพัฒนาโดยคำนึงถึงประสบการณ์ของผู้ใช้เป็นหลัก ทำให้ระบบตอบโจทย์ความต้องการได้ดียิ่งขึ้น

### 2.3.6 Usability Testing

Usability Testing เป็นกระบวนการทดสอบเพื่อประเมินว่าเว็บแอปพลิเคชัน ScoreOBE+ สามารถใช้งานได้ง่ายและมีประสิทธิภาพเพียงใด โดยทดสอบกับผู้ใช้จริงเพื่อตรวจสอบปัญหาที่อาจเกิดขึ้น และนำข้อมูลที่ได้มาปรับปรุงระบบให้ตอบสนองความต้องการของผู้ใช้มากขึ้น

#### วัตถุประสงค์ของ Usability Testing

- ตรวจสอบว่าผู้ใช้สามารถทำงานในระบบได้อย่างราบรื่นหรือไม่
- ระบุปัญหาด้านการใช้งานที่อาจเป็นอุปสรรคต่อผู้ใช้
- ปรับปรุงประสบการณ์ผู้ใช้ (User Experience) เพื่อให้ใช้งานได้ง่ายขึ้น
- วิเคราะห์เวลาและขั้นตอนที่ใช้ในการดำเนินงานภายในระบบ

#### กระบวนการ Usability Testing

- กำหนดเป้าหมายการทดสอบ - กำหนดวัตถุประสงค์ เช่น การทดสอบการใช้งานฟังก์ชันการอัปโหลด คะแนน หรือการถ่ายภาพ
- เลือกกลุ่มตัวอย่างผู้ใช้ - คัดเลือกผู้ใช้ที่มีบทบาทต่าง ๆ ได้แก่ อาจารย์ นักศึกษา และเจ้าหน้าที่ฝ่ายวิชาการ เพื่อทดสอบระบบในมุมมองที่หลากหลาย
- เตรียมชุดทดสอบ (Test Scenarios) - ออกแบบชุดคำถามหรือสถานการณ์เพื่อให้ผู้ใช้ทำงานผ่านระบบ เช่น การเพิ่มคะแนน การถ่ายภาพ และการถ่ายภาพ
- ดำเนินการทดสอบ - ให้ผู้ใช้ทดลองใช้งานจริงโดยไม่ให้คำแนะนำเพิ่มเติม และเก็บข้อมูลพฤติกรรมของผู้ใช้ระหว่างการทดสอบ
- วิเคราะห์ข้อมูลและสรุปผล - วิเคราะห์ข้อผิดพลาด ปัญหาการใช้งาน และรวมข้อเสนอแนะจากผู้ใช้ เพื่อนำไปพัฒนาระบบที่เหมาะสมยิ่งขึ้น

#### เครื่องมือที่ใช้ในการทดสอบ

- ระบบบันทึกวิดีโอบนจอ (Screen Recording) เพื่อตรวจสอบพฤติกรรมการใช้งาน
- แบบสอบถาม (Survey) เพื่อรับรวมความคิดเห็นของผู้ใช้

#### ผลลัพธ์ที่คาดหวังจาก Usability Testing

- ระบบสามารถใช้งานได้ง่ายขึ้น และลดปัญหาด้าน UX/UI
- ผู้ใช้สามารถเข้าใจวิธีการใช้งานโดยไม่ต้องมีการอธิบายเพิ่มเติม
- ปรับปรุงประสิทธิภาพของระบบให้เหมาะสมกับการใช้งานจริง
- เพิ่มอัตราการยอมรับและความพึงพอใจของผู้ใช้

### 2.3.7 Feedback Capture Grid

Feedback Capture Grid เป็นเครื่องมือที่ใช้ในการรวบรวมความคิดเห็นของผู้ใช้ระหว่างการทดสอบการใช้งาน (**Usability Testing**) โดยช่วยให้ทีมพัฒนาสามารถเข้าใจจุดแข็ง จุดอ่อน และโอกาสในการปรับปรุงระบบได้อย่างเป็นระบบ เครื่องมือนี้แบ่งความคิดเห็นของผู้ใช้ออกเป็น 4 หมวดหมู่หลัก ได้แก่

1. **สิ่งที่ชอบ (Likes)** หมวดหมู่นี้ใช้รวบรวมจุดที่ผู้ใช้พึงพอใจและชื่นชอบเกี่ยวกับระบบ ซึ่งเป็นองค์ประกอบที่ช่วยให้ประสบการณ์ใช้งานเป็นไปในทางบวก

- อินเทอร์เฟชที่ใช้งานง่าย สวยงาม มินิมอล และมีโครงสร้างชัดเจน
- กราฟแสดงผลคะแนนที่อ่านง่าย และมีการเน้นจุดสำคัญ
- ระบบแจ้งเตือนเมื่อคะแนนมีการอัปเดต

ตัวอย่างจาก **ScoreOBE+**:

- “ชอบที่สามารถดูคะแนนย้อนหลังได้ง่าย ไม่ต้องค้นหาหลายขั้นตอน”
- “หน้าจอแสดงคะแนนเข้าใจง่าย ดูแล้วรู้ทันทีว่าคะแนนแต่ละส่วนเป็นอย่างไร”
- “ชอบที่สามารถใช้ทั้งบนมือถือและคอมพิวเตอร์ได้ โดยอินเทอร์เฟชปรับให้เหมาะสมกับขนาดหน้าจอ”

2. **สิ่งที่ต้องปรับปรุง (Criticisms)** หมวดหมู่นี้ใช้รวบรวมข้อบกพร่องหรือปัญหาที่ทำให้ผู้ใช้ไม่พึงพอใจ และเป็นสิ่งที่ควรได้รับการปรับปรุงเพื่อเพิ่มประสิทธิภาพของระบบ

- การเปลี่ยนไปมาระหว่างหน้า อาจต้องกดหลายครั้ง
- สีของกราฟบางสีคล้ายกันมาก ทำให้แยกความแตกต่างได้ยาก
- ไม่มีฟังก์ชันเปรียบเทียบคะแนนของตนเองกับค่าเฉลี่ยของกลุ่ม
- ระบบไม่แจ้งเตือนเมื่อมีการเปลี่ยนแปลงคะแนนจากอาจารย์

ตัวอย่างจาก **ScoreOBE+**:

- “ถ้าระบบสามารถให้ผู้ใช้กำหนดสีของกราฟเองได้ น่าจะช่วยให้แยกแต่ละหมวดหมู่คะแนนได้ง่ายขึ้น”
- “สีไม่สด อย่างได้สีที่ดูน่าสดใส แต่ดูโว้กมากกว่านี้”

3. **ไอเดียใหม่ (Ideas)** หมวดหมู่นี้ใช้รวบรวมข้อเสนอแนะหรือแนวคิดใหม่ ๆ จากผู้ใช้ที่อาจช่วยปรับปรุงประสบการณ์การใช้งาน หรือเพิ่มฟีเจอร์ที่ตอบโจทย์ความต้องการของผู้ใช้มากขึ้น

- ควรเพิ่มโหมดมืด (Dark Mode) เพื่อให้สามารถใช้งานได้สบายตาขึ้นในที่แสงน้อย
- ระบบสามารถให้ผู้ใช้ตั้งค่าเป้าหมายคะแนนของตัวเองได้ เพื่อช่วยกระตุ้นการเรียน
- ควรมีการสรุปคะแนนโดยอัตโนมัติเป็น Dashboard ที่แสดงข้อมูลสำคัญทั้งหมดในหน้าเดียว

- ควรมีระบบเปรียบเทียบคคะแนนกับเพื่อนในกระบวนวิชาเดียวกัน เพื่อให้เห็นแนวโน้มของกลุ่ม

ตัวอย่างจาก **ScoreOBE+:**

- “ถ้ามีระบบแจ้งเตือนเมื่อคะแนนเปลี่ยนแปลง หรือคะแนนต่ำกว่าค่าเฉลี่ยของกลุ่ม”
- “อย่างให้สามารถเลือกดูคะแนนของแต่ละกระบวนวิชาแบบเรียงตามลำดับคะแนนสูงไปต่ำ หรือจากต่ำไปสูงได้”

4. คำถาม (**Questions**) หมวดหมู่นี้ใช้รวบรวมคำถามที่เกิดขึ้นระหว่างการทดสอบ ซึ่งอาจเป็นข้อสงสัย เกี่ยวกับฟังก์ชันของระบบ หรือข้อสงสัยที่อาจนำไปสู่การพัฒนาเพิ่มเติม

- สามารถดูคะแนนของทุกวิชาแบบรวมกันในหน้าเดียวได้ไหม?
- การคำนวณคะแนนรวมมีหลักเกณฑ์อะไรบ้าง? สามารถเปลี่ยนการคำนวณให้สอดคล้องกับหลักสูตรได้หรือไม่?
- นักศึกษาสามารถส่งคำร้องขอให้แก้ไขคะแนนได้หรือไม่?

ตัวอย่างจาก **ScoreOBE+:**

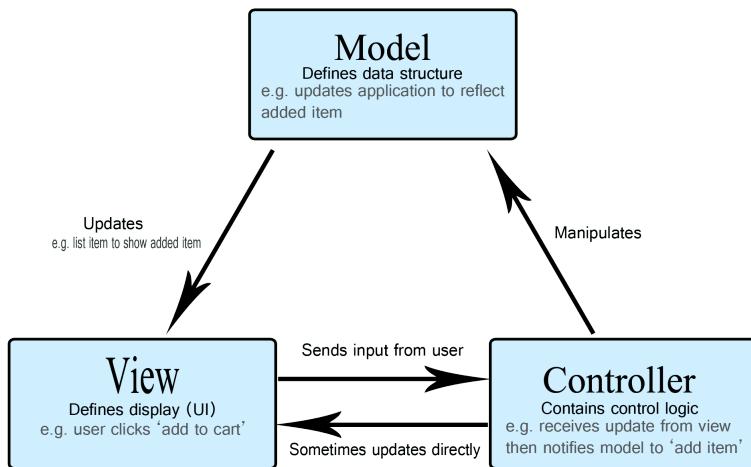
- “สามารถเปรียบเทียบคคะแนนของตัวเองกับเพื่อนในคลาสได้โดยไม่ต้องให้เพื่อนแชร์คะแนน เองหรือไม่”
- “การแสดงผลคะแนนในรูปแบบกราฟมีการอัปเดตอัตโนมัติเมื่อคะแนนเปลี่ยนแปลงหรือไม่”

## 2.4 Platform Development

### 2.4.1 MVC

MVC ย่อมาจาก Model-View-Controller เป็น Software Design Pattern แบ่งออกเป็น 3 ส่วน [17]

1. **M (Model)** คือ ส่วนที่เกี่ยวข้องกับการจัดการข้อมูลหรือ Business logic
2. **V (View)** คือ ส่วนที่เกี่ยวข้องกับ User Interface
3. **C (Controller)** คือ ส่วนที่เป็นส่วนการจัดการ request และ response ของระบบ หรือเป็นตัวที่เชื่อมโยงระหว่าง Model และ View



รูปที่ 2.5: Model-View-Controller

#### 2.4.2 Rest API

REST ย่อมาจาก Representational State Transfer เป็นรูปแบบการส่งข้อมูลระหว่าง Server-Client รูปแบบหนึ่งซึ่งอยู่บนพื้นฐานของ HTTP Protocol เป็นการสร้าง Web Service เพื่อแลกเปลี่ยนข้อมูลกันผ่าน Application วิธีหนึ่ง ซึ่งส่งข้อมูลได้หลายชนิด ไม่ว่าจะเป็น Text, XML, JSON หรือแม้แต่ HTML ก็สามารถส่งได้ [27]

API ย่อมาจาก Application Programming Interface คือการเชื่อมต่อจากระบบหนึ่งไปสู่อีกระบบหนึ่ง เพื่อให้ซอฟต์แวร์ภายนอกเข้าถึงและอัปเดตข้อมูลนั้นๆได้ แต่ยังอยู่ในขอบเขตที่ถูกกำหนดไว้ กล่าวคือ API เป็นตัวกลางที่จะทำให้คุณรับคำสั่งต่าง ๆ ประมวลผลและกระทำการข้อมูลส่งกลับคืนไปยังคุณสั่งโดยอัตโนมัติ [1]

#### 2.4.3 Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) เป็นแนวทางในการควบคุมการเข้าถึงระบบตามบทบาท (Roles) ของผู้ใช้ โดยผู้ใช้แต่ละคนจะได้รับสิทธิ์ในการเข้าถึงข้อมูลและฟังก์ชันต่าง ๆ ตามบทบาทที่ได้รับมอบหมาย ระบบนี้ช่วยเพิ่มความปลอดภัยและลดความซับซ้อนในการจัดการสิทธิ์ของผู้ใช้ภายในระบบ [?]

#### องค์ประกอบหลักของ RBAC

RBAC ประกอบด้วยองค์ประกอบหลักดังต่อไปนี้:

1. **ผู้ใช้ (Users)** หมายถึงบุคคลที่ใช้ระบบ และได้รับสิทธิ์การเข้าถึงตามบทบาทที่ได้รับ ตัวอย่างเช่น นักศึกษา อาจารย์ หรือผู้ดูแลระบบใน ScoreOBE+
2. **บทบาท (Roles)** หมายถึงกลุ่มของสิทธิ์ที่กำหนดให้กับผู้ใช้แต่ละประเภท ซึ่งแต่ละบทบาทสามารถเข้าถึงและดำเนินการในระบบได้แตกต่างกัน

3. สิทธิ์ (Permissions) หมายถึงการอนุญาตให้ดำเนินการบางอย่างในระบบ เช่น การดูข้อมูล แก้ไข ลบ หรืออัปโหลดค่าแนว
4. ทรัพยากร (Resources) หมายถึงข้อมูลหรือฟังก์ชันที่สามารถเข้าถึงได้ เช่น ฐานข้อมูลค่าแนว หน้า การจัดการผู้ใช้ หรือโมดูลสถิติ

### ตัวอย่าง RBAC ใน ScoreOBE+

ในระบบ ScoreOBE+ มีการกำหนดบทบาทและสิทธิ์การเข้าถึงของผู้ใช้แต่ละประเภท ดังนี้:

Role	View Scores	Upload Scores	Edit Scores	Manage Users
นักศึกษา (Student)	✓	-	-	-
อาจารย์ (Instructor)	✓	✓	✓	-
ผู้ดูแลระบบ (Admin)	✓	✓	✓	✓

ตารางที่ 2.1: การกำหนดสิทธิ์การเข้าถึงตามบทบาทใน ScoreOBE+

### ตัวอย่างการใช้งาน RBAC ในระบบ ScoreOBE+:

- นักศึกษา (Student) สามารถดูค่าแนวของตนเองได้เท่านั้น แต่ไม่สามารถแก้ไขหรืออัปโหลดค่าแนวได้
- อาจารย์ (Instructor) สามารถอัปโหลดและแก้ไขค่าแนวของนักศึกษาในกระบวนวิชาที่สอน แต่ไม่สามารถจัดการบัญชีผู้ใช้ได้
- ผู้ดูแลระบบ (Admin) มีสิทธิ์เข้าถึงทุกฟังก์ชัน รวมถึงการจัดการบัญชีผู้ใช้และสิทธิ์การเข้าถึงของแต่ละบทบาท

### ข้อดีของ RBAC

RBAC มีข้อดีหลายประการที่ช่วยเพิ่มประสิทธิภาพและความปลอดภัยในการจัดการสิทธิ์ของผู้ใช้ ดังนี้:

- ความปลอดภัยสูงขึ้น (Enhanced Security) RBAC ช่วยลดความเสี่ยงของการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต โดยจำกัดสิทธิ์ของผู้ใช้ตามบทบาทที่ได้รับ
- ลดความซับซ้อนในการจัดการสิทธิ์ (Simplified Permission Management) เนื่องจากสิทธิ์ถูกกำหนดตามบทบาท การเพิ่ม หรือลดสิทธิ์ของผู้ใช้สามารถทำได้ง่ายโดยการเปลี่ยนบทบาทแทนการแก้ไขสิทธิ์ทีละรายการ
- เพิ่มประสิทธิภาพของระบบ (Improved System Efficiency) ระบบสามารถควบคุมการเข้าถึงข้อมูลได้อย่างเหมาะสม ลดภาระของผู้ดูแลระบบในการกำหนดสิทธิ์ให้ผู้ใช้เป็นรายบุคคล
- รองรับมาตรฐานด้านความปลอดภัย (Compliance with Security Standards) RBAC เป็นแนวทางที่ได้รับการยอมรับและใช้ในมาตรฐานความปลอดภัยต่าง ๆ เช่น ISO 27001 และ NIST

## ข้อควรพิจารณาในการใช้ RBAC

แม้ว่า RBAC จะมีข้อดีมามาย แต่ก็มีข้อควรพิจารณาที่ต้องคำนึงถึง เช่น:

- การออกแบบบทบาทต้องคำนึงถึงความยืดหยุ่น และไม่ควรซับซ้อนเกินไป
- ต้องมีการบริหารจัดการบทบาทและสิทธิอย่างต่อเนื่อง เพื่อให้สอดคล้องกับการเปลี่ยนแปลงของระบบ
- การกำหนดบทบาทที่ไม่เหมาะสมอาจทำให้เกิดความยุ่งยากในการควบคุมสิทธิ์ของผู้ใช้

### 2.4.4 JSON Web Token (JWT)

JSON Web Token (JWT) เป็นมาตรฐานเปิด (*open standard*) ที่ใช้สำหรับการรับรองตัวตน (*authentication*) และการแลกเปลี่ยนข้อมูล (*information exchange*) ระหว่างฝ่ายที่เกี่ยวข้อง โดยมีการเข้ารหัสเพื่อให้มั่นใจว่าข้อมูลที่ถูกส่งไม่ถูกแก้ไขระหว่างทาง [?]

#### โครงสร้างของ JWT

JWT ประกอบด้วยสามส่วนหลักที่ถูกเข้ารหัสด้วยอัลกอริธึมเข้ารหัส เช่น HMAC หรือ RSA และถูกคั่นด้วยเครื่องหมาย . ดังนี้:

Header.Payload.Signature

1. **Header** เป็นส่วนที่ระบุประเภทของโทเคนและอัลกอริธึมที่ใช้เข้ารหัส ตัวอย่างของ Header ใน JWT:

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

2. **Payload** เป็นส่วนที่เก็บข้อมูล (*claims*) ซึ่งอาจมีข้อมูลเกี่ยวกับผู้ใช้ เช่น sub (subject), name, และ exp (expiration time) ตัวอย่างของ Payload:

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "iat": 1710867200  
}
```

3. **Signature** ใช้สำหรับตรวจสอบความถูกต้องของโทเคน โดยทำการเข้ารหัสข้อมูล Header และ Payload ด้วยคีย์ลับและอัลกอริธึมที่กำหนด เช่น HMAC-SHA256

## JWT ใน ScoreOBE+

ในระบบ ScoreOBE+ JWT ถูกนำมาใช้เพื่อจัดการการรับรองตัวตนของผู้ใช้และการควบคุมสิทธิ์การเข้าถึง (*authorization*) ดังนี้:

- เมื่อผู้ใช้เข้าสู่ระบบ ระบบจะสร้าง JWT และส่งให้ผู้ใช้เก็บไว้ในเบราว์เซอร์ (เช่น Local Storage หรือ HTTP Cookie)
- เมื่อมีการร้องขอ API ที่ต้องมีการยืนยันตัวตน JWT จะถูกส่งไปใน Authorization Header เช่น:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

- ผู้ใช้ที่รับ JWT จะเช็คJWT ตรวจสอบความถูกต้องของโทเค็น และอนุญาตให้เข้าถึงทรัพยากรที่ร้องขอหาก JWT ถูกต้อง

## ข้อดีของ JWT

JWT มีข้อดีหลายประการที่ทำให้เป็นทางเลือกยอดนิยมในการจัดการการรับรองตัวตน:

- ไม่ต้องพึ่งพาฐานข้อมูลทุกครั้ง (Stateless Authentication)** JWT เก็บข้อมูลภายในตัวโทเค็น ทำให้เซิร์ฟเวอร์ไม่ต้องตรวจสอบฐานข้อมูลทุกครั้งที่มีการร้องขอ
- รองรับการใช้งานข้ามโดเมน (Cross-Domain Authentication)** สามารถใช้ JWT ร่วมกับ OAuth 2.0 เพื่อให้บริการภายนอกสามารถยืนยันตัวตนได้
- มีโครงสร้างที่เล็กและสามารถอ่านได้ง่าย** ข้อมูลใน JWT ถูกเข้ารหัสแบบ Base64 ทำให้สามารถอ่านได้โดยทั่วไป
- รองรับการกำหนดเวลาใช้งาน (Token Expiration)** JWT สามารถกำหนดวันหมดอายุเพื่อเพิ่มความปลอดภัย เช่น การกำหนด exp ให้โทเค็นหมดอายุภายใน 1 ชั่วโมง

## ข้อควรระวังในการใช้ JWT

แม้ว่า JWT จะมีข้อดีมากมาย แต่ก็มีข้อควรระวังที่ต้องคำนึงถึง:

- หากใช้ JWT แบบไม่มีการเข้ารหัส (Plaintext) ข้อมูลอาจถูกเปิดเผยได้
- หากใช้โทเค็นแบบปั๊มมีการหมดอายุ (Permanent Token) อาจทำให้เกิดปัญหาด้านความปลอดภัย
- การเก็บ JWT ใน Local Storage อาจทำให้ถูกขโมยผ่านการโจมตีแบบ XSS (Cross-Site Scripting)

## 2.4.5 Containerization (Docker)

Containerization เป็นเทคนิคในการพัฒนาและใช้งานซอฟต์แวร์ที่ช่วยให้ออปพลิเคชันสามารถรันได้ในสภาพแวดล้อมที่แยกออกจากกัน โดยใช้แนวคิดของคอนเทนเนอร์ (*Container*) ซึ่งมีความคล้ายคลึงกับ Virtual Machine (VM) แต่ใช้ทรัพยากรที่มีประสิทธิภาพมากกว่า

### Docker คืออะไร?

Docker เป็นแพลตฟอร์มสำหรับการจัดการคอนเทนเนอร์ที่ช่วยให้ผู้พัฒนาสามารถสร้าง แพ็กเกจ และรัน ออปพลิเคชันในคอนเทนเนอร์ โดย Docker ช่วยลดปัญหาความแตกต่างของสภาพแวดล้อมระหว่างเครื่อง พัฒนาและเครื่องจริงเซิร์ฟเวอร์

### โครงสร้างของ Docker

Docker มีองค์ประกอบสำคัญดังต่อไปนี้:

- **Docker Image** ไฟล์ต้นแบบที่บรรจุได้ด้วย ออปพลิเคชัน ไลบรารี และคุณลักษณะที่จำเป็นเพื่อรันคอนเทนเนอร์
- **Docker Container** หน่วยที่ทำงานจริงของออปพลิเคชัน ซึ่งถูกสร้างขึ้นจาก Docker Image
- **Dockerfile** ไฟล์ที่ใช้กำหนดวิธีการสร้าง Docker Image โดยสามารถบุคคลสั่ง เช่น การติดตั้งแพ็กเกจ การคัดลอกไฟล์ และการกำหนดค่าต่างๆ
- **Docker Compose** เครื่องมือที่ช่วยจัดการหลายคอนเทนเนอร์พร้อมกันผ่านไฟล์ `docker-compose.yml`
- **Docker Hub** แหล่งเก็บและแชร์ Docker Image แบบออนไลน์ ซึ่งช่วยให้สามารถดึงอิมเมจมาใช้งานได้

### การทำงานของ Docker

Docker ทำงานโดยใช้ *Docker Engine* ซึ่งเป็นตัวกลางที่จัดการการสร้างและรันคอนเทนเนอร์ โดยมีลำดับการทำงานดังนี้:

1. นักพัฒนาเขียน Dockerfile และใช้คำสั่ง `docker build` เพื่อสร้าง Docker Image
2. ใช้คำสั่ง `docker run` เพื่อสร้างและรัน Docker Container จาก Image
3. คอนเทนเนอร์ทำงานในสภาพแวดล้อมที่แยกออกจากโสสต์ แต่สามารถสื่อสารผ่านเครือข่ายที่กำหนด
4. สามารถใช้ `docker-compose up` เพื่อรันหลายคอนเทนเนอร์พร้อมกัน

## การใช้ Docker ใน ScoreOBE+

Docker ถูกนำมาใช้ในระบบ ScoreOBE+ เพื่อช่วยให้การพัฒนาและการใช้งานระบบง่ายขึ้น โดยมีการใช้งานดังนี้:

- การพัฒนาแยกสภาพแวดล้อม ใช้ Docker เพื่อสร้างสภาพแวดล้อมพัฒนาให้เหมือนกับเซิร์ฟเวอร์จริง ลดปัญหาความแตกต่างระหว่างเครื่องพัฒนา
- การรัน **Backend** และ **Database** ในคอนเทนเนอร์ ScoreOBE+ ใช้ Docker Compose เพื่อจัดการคอนเทนเนอร์ของ backend และฐานข้อมูล MongoDB
- การปรับใช้ (**Deployment**) ที่ง่ายขึ้น สามารถแพ็คระบบทั้งหมดเป็น Docker Image และนำไปใช้งานบนเซิร์ฟเวอร์ได้

### ข้อดีของ Docker

- Portability** สามารถรันแอปพลิเคชันได้บนทุกแพลตฟอร์มที่รองรับ Docker โดยไม่ต้องตั้งค่าใหม่
- การใช้ทรัพยากรอย่างมีประสิทธิภาพ ใช้ทรัพยากรน้อยกว่า Virtual Machine เพราะใช้เครื่องในล่องโซลิสต์ร่วมกัน
- ปรับปรุงกระบวนการ **CI/CD** Docker สามารถช่วยให้พัฒนาทดสอบและปรับใช้ซอฟต์แวร์ได้เร็ว

### ข้อควรระวังในการใช้ Docker

- คอนเทนเนอร์อาจมีปัญหาด้านความปลอดภัยหากไม่ได้กำหนดสิทธิ์การเข้าถึงอย่างเหมาะสม
- การใช้ทรัพยากรของโซลิสต์อาจสูงขึ้นหากมีการรันคอนเทนเนอร์จำนวนมาก
- คอนเทนเนอร์ไม่สามารถแทนที่ Virtual Machine ได้ในทุกรูปแบบ โดยเฉพาะแอปพลิเคชันที่ต้องการระบบปฏิบัติการเดิมรูปแบบ

### 2.4.6 Separation of Concerns (SoC)

หลักการ **Separation of Concerns (SoC)** เป็นแนวคิดทางวิศวกรรมซอฟต์แวร์ที่มุ่งเน้นการแยกส่วนของระบบออกเป็นโมดูลหรือองค์ประกอบที่มีหน้าที่ชัดเจนและไม่ทับซ้อนกัน หลักการนี้ช่วยให้โค้ดมีโครงสร้างที่ดีขึ้น ลดความซับซ้อน และเพิ่มความสามารถในการดูแลรักษา (Maintainability)

#### แนวคิดของ Separation of Concerns

หลักการ SoC มีพื้นฐานมาจากแนวคิดที่ว่า ”โค้ดที่จัดการเรื่องหนึ่งไม่ควรยุ่งเกี่ยวกับเรื่องอื่น” ซึ่งหมายความว่าแต่ละส่วนของระบบควรรับผิดชอบเพียงด้านเดียว ตัวอย่างของการใช้หลักการนี้ได้แก่:

- การแยกส่วนระหว่าง **Frontend** และ **Backend**
- การแบ่งโครงสร้างซอฟต์แวร์เป็นレイเยอร์ เช่น **MVC (Model-View-Controller)**
- การแยกส่วน **API**, ฐานข้อมูล และการประมวลผลธุรกิจ

## ตัวอย่างการใช้งานใน ScoreOBE+

ระบบ ScoreOBE+ ใช้แนวคิด Separation of Concerns เพื่อให้โค้ดมีโครงสร้างที่เป็นระเบียบและง่ายต่อการดูแล โดยมีการแยกส่วนดังนี้:

- **Backend (NestJS)** - จัดการกับ API และตระกูลของระบบ เช่น การคำนวณคะแนน และการจัดเก็บข้อมูลในฐานข้อมูล
- **Frontend (React)** - แสดงผลข้อมูลให้กับผู้ใช้ และจัดการการโต้ตอบ (User Interaction)
- **Database (MongoDB)** - จัดเก็บข้อมูลของนักศึกษา และข้อมูลกระบวนการวิชา

## ข้อดีของ Separation of Concerns

- เพิ่มความสามารถในการดูแลรักษา (Maintainability) - นักพัฒนาสามารถปรับปรุงหรือแก้ไขบางส่วนของระบบโดยไม่ส่งผลกระทบต่อส่วนอื่น
- เพิ่มความสามารถในการขยายระบบ (Scalability) - แต่ละโมดูลสามารถถูกพัฒนาให้ขึ้นชั้นๆ ได้โดยไม่กระทบระบบทั้งหมด
- ลดความซับซ้อนของโค้ด (Code Simplicity) - โค้ดมีโครงสร้างที่อ่านง่ายและแยกเป็นสัดส่วน

## ตัวอย่างแนวทางปฏิบัติที่ดี (Best Practices)

- ใช้โครงสร้าง **MVC (Model-View-Controller)** หรือ **MVVM (Model-View-ViewModel)**
- แยกโค้ด API, และการจัดเก็บข้อมูลออกจากกัน
- ใช้ Microservices หรือ Modular Design เพื่อลดการพึ่งพาภันระหว่างโมดูล

### 2.4.7 Modular Architecture

Modular Architecture เป็นแนวคิดการออกแบบระบบที่แบ่งโครงสร้างของซอฟต์แวร์ออกเป็นโมดูลย่อย ๆ (Modules) ซึ่งทำงานแยกจากกัน แต่สามารถทำงานร่วมกันได้ผ่าน API หรืออินเตอร์เฟซที่กำหนดไว้ โมดูลเหล่านี้สามารถพัฒนา ทดสอบ และปรับปรุงได้โดยไม่ส่งผลกระทบต่อส่วนอื่นของระบบ

## โครงสร้างโมดูลของ ScoreOBE+

ระบบ ScoreOBE+ ถูกออกแบบให้เป็น Modular Architecture โดยแบ่งออกเป็นโมดูลหลัก ๆ ดังนี้:

- **Frontend Module:** พัฒนาโดยใช้ *React* ทำหน้าที่เป็นอินเทอร์เฟซสำหรับอาจารย์และนักศึกษา ในการดูแลและการจัดการข้อมูลเกี่ยวกับคะแนน
- **Backend Module:** ใช้ *NestJS* สำหรับสร้าง API และจัดการข้อมูลที่ส่งมาจาก Frontend รวมถึงการตรวจสอบสิทธิ์ของผู้ใช้
- **Authentication Module:** ใช้ *JSON Web Token (JWT)* และ *Role-Based Access Control (RBAC)* ในการจัดการสิทธิ์ของผู้ใช้แต่ละประเภท เช่น อาจารย์ นักศึกษา และผู้ดูแลระบบ

- **Database Module:** ใช้ *MongoDB* ในการจัดเก็บข้อมูลเกี่ยวกับกระบวนการวิชา นักศึกษา คะแนน และผลการเรียนรู้ตามแนวคิด OBE
- **File Processing Module:** รองรับการอัปโหลดไฟล์คณิตในรูปแบบ *CSV/XLSX* และตรวจสอบข้อมูลพลาดในไฟล์อัตโนมัติ
- **Analytics Module:** ใช้สำหรับคำนวณค่าสถิติต่าง ๆ เช่น ค่าเฉลี่ยของคะแนนนักศึกษา ค่าความแปรปรวน และแนวโน้มผลการเรียน

### ข้อดีของ Modular Architecture

- **แยกส่วนการพัฒนาได้ง่าย:** ผู้พัฒนาสามารถทำงานกับแต่ละโมดูลได้อย่าง ทำให้กระบวนการพัฒนาเร็วมากขึ้น
- **รองรับการขยายระบบ:** สามารถเพิ่มโมดูลใหม่ได้โดยไม่ต้องแก้ไขโครงสร้างหลักของระบบ
- **การทดสอบและบำรุงรักษาง่าย:** สามารถทดสอบโมดูลแยกกันได้ (Unit Testing) และหากเกิดปัญหา กับโมดูลใดโมดูลหนึ่ง ก็สามารถแก้ไขได้โดยไม่กระทบกับระบบทั้งหมด
- **ปรับเปลี่ยนเทคโนโลยีได้ง่าย:** หากต้องการเปลี่ยนแปลงเทคโนโลยีของโมดูลใด ๆ เช่น เปลี่ยนจาก *NestJS* เป็น *Express* ใน Backend ก็สามารถทำได้โดยไม่กระทบต่อโมดูลอื่น

## 2.5 Database

ฐานข้อมูลที่ใช้ในโครงการนี้ คือ *MongoDB* เป็นระบบฐานข้อมูล NoSQL ซึ่งจัดเก็บและดึงข้อมูลด้วยวิธีที่ยืดหยุ่นและปรับขนาดได้ เก็บข้อมูลในรูปแบบของเอกสารคล้าย *JSON* พร้อม *schema* โดยการจัดเก็บข้อมูลจะเก็บเป็นชุด (*Collection*) คล้ายกับตารางในฐานข้อมูลแบบ *SQL* โดยเอกสารแต่ละฉบับในชุดสามารถมีโครงสร้างที่แตกต่างกัน ทำให้ประเภทข้อมูลที่สามารถจัดเก็บมีความยืดหยุ่นมากขึ้น [14]

## 2.6 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ

1. 261207 - Basic Computer Engineering Laboratory การพัฒนาเว็บแอปพลิเคชัน
2. 261342 - Fundamentals of Database Systems การออกแบบและจัดการฐานข้อมูล
3. 261497 - Full Stack Development (1/66) การ deploy เว็บแอปพลิเคชัน

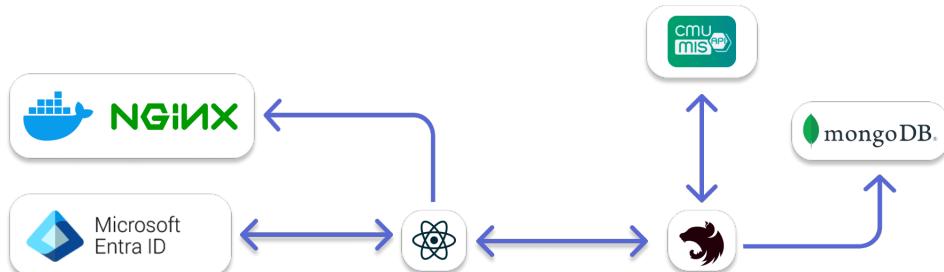
## 2.7 ความรู้อกหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ

1. การออกแบบ *User interface* และ *User experience*
2. การใช้งาน *เฟรมเวิร์ก* และ *ไลบรารี*ต่าง ๆ สำหรับการทำเว็บแอปพลิเคชัน

## บทที่ 3

### โครงสร้างและขั้นตอนการทำงาน

#### 3.1 สถาปัตยกรรมระบบ



รูปที่ 3.1: System Diagram

ระบบนี้ประกอบไปด้วยองค์ประกอบหลัก ได้แก่

##### 3.1.1 Authentication

ในระบบ ScoreOBE+ การยืนยันตัวตนของผู้ใช้ (Authentication) เป็นส่วนสำคัญที่ช่วยให้ผู้ใช้แต่ละคนสามารถเข้าถึงข้อมูลและฟังก์ชันที่เหมาะสมกับบทบาทของตน ระบบนี้ใช้ **Microsoft Entra ID** (เดิมคือ *Azure Active Directory* หรือ *Azure AD*) เป็นโซลูชันหลักสำหรับการจัดการการตรวจสอบสิทธิ์ของผู้ใช้ โดยผู้ใช้สามารถเข้าสู่ระบบผ่าน **CMU Account** ซึ่งเป็นบัญชีที่ใช้ภายในมหาวิทยาลัยเชียงใหม่

##### การทำงานของ Authentication

เมื่อผู้ใช้ต้องการเข้าสู่ระบบ ScoreOBE+ ระบบจะทำการตรวจสอบสิทธิ์โดยใช้ Microsoft Entra ID ผ่านขั้นตอนดังนี้:

- ผู้ใช้เข้าสู่หน้าเข้าสู่ระบบ (**Login Page**) เมื่อผู้ใช้ต้องการเข้าสู่ระบบ ScoreOBE+ ระบบจะเปลี่ยนเส้นทางไปยังหน้าล็อกอินของ Microsoft Entra ID เพื่อให้ผู้ใช้ป้อนข้อมูลรับรอง (CMU Account)
- ตรวจสอบสิทธิ์ด้วย **OAuth 2.0** และ **OpenID Connect** Microsoft Entra ID ใช้ **OAuth 2.0** และ **OpenID Connect** ในการตรวจสอบตัวตนของผู้ใช้ หลังจากที่ผู้ใช้กรอกข้อมูลรับรอง (Username และ Password) ระบบจะตรวจสอบความถูกต้องของบัญชีและสิทธิ์การเข้าถึง
- ออกโทเคนยืนยันตัวตน (**Access Token & JWT Token**) หากการตรวจสอบสำเร็จ Microsoft Entra ID จะออก **Access Token** ให้กับผู้ใช้ ซึ่งระบบจะใช้เพื่อตรวจสอบสิทธิ์และดึงข้อมูลบัญชีจาก Entra ID จากนั้น Backend จะใช้ข้อมูลที่ได้รับสร้าง **JWT Token** ผ่าน NestJS Passport และ **JWT** เพื่อใช้ในการตรวจสอบสิทธิ์ในระบบ ScoreOBE+ โดย

- **Access Token** (จาก Entra ID) ใช้สำหรับเข้าถึง API ของ Microsoft Entra ID เพื่อดึงข้อมูลบัญชี
  - **JWT Token** (จาก Backend) ใช้สำหรับจัดการเรสซันและยืนยันตัวตนของผู้ใช้ภายใน ScoreOBE+ โดยจะถูกแนบไปกับทุกคำขอของระบบ เพื่อตรวจสอบความถูกต้องของ Token และให้สิทธิ์เข้าถึงตามบทบาทของผู้ใช้
4. ระบบตรวจสอบบทบาทของผู้ใช้ (**Role-Based Access Control - RBAC**) เมื่อได้รับโทคেนแล้ว ระบบ ScoreOBE+ จะทำการตรวจสอบบทบาทของผู้ใช้ ( เช่น อาจารย์ นักศึกษา หรือผู้ดูแลระบบ) ผ่านค่าที่อยู่ในโทคีน เพื่อกำหนดสิทธิ์การเข้าถึงข้อมูลและฟังก์ชันต่าง ๆ ในระบบ
5. ให้สิทธิ์การเข้าถึงระบบ **ScoreOBE+** หากการตรวจสอบสิทธิ์สำเร็จ ผู้ใช้สามารถเข้าถึง ScoreOBE+ และใช้งานฟีเจอร์ต่าง ๆ ได้ตามบทบาทที่ได้รับ เช่น อาจารย์สามารถอัปโหลดคะแนน นักศึกษาสามารถดูผลการเรียน และผู้ดูแลระบบสามารถจัดการข้อมูลระบบ

### 3.1.2 Frontend

Frontend ของ ScoreOBE+ ถูกพัฒนาขึ้นโดยใช้ **React** ซึ่งเป็น JavaScript Library สำหรับสร้าง **User Interface (UI)** React ช่วยให้สามารถพัฒนา UI แบบ **Component-Based** ทำให้โค้ดมีโครงสร้างที่เป็นระบบและสามารถนำกลับมาใช้ซ้ำได้ นอกจากนี้ยังมีการใช้งาน **State Management** ผ่าน Redux เพื่อให้การจัดการข้อมูลระหว่าง Component

#### หน้าที่หลักของ Frontend

Frontend ใน ScoreOBE+ มีหน้าที่สำคัญในการให้ผู้ใช้สามารถโต้ตอบกับระบบได้ โดยมีบทบาทหลักดังต่อไปนี้:

- การยืนยันตัวตนและการเข้าสู่ระบบ
 

ระบบใช้ **Microsoft Entra ID** (เดิมคือ Azure AD) สำหรับจัดการ Authentication ผ่าน **OAuth 2.0** และ **OpenID Connect** โดยผู้ใช้สามารถเข้าสู่ระบบด้วย **CMU Account**

#### การสื่อสารกับ Backend

เมื่อผู้ใช้ดำเนินการใด ๆ ที่ต้องมีการดึงข้อมูล เช่น การดูคะแนน หรือการอัปโหลดไฟล์ ระบบ Frontend จะส่ง **HTTP Request** ไปยัง Backend ผ่าน **RESTful API** โดยใช้ไลบรารี เช่น axios

#### การแสดงผลข้อมูลให้ผู้ใช้

ข้อมูลที่ได้รับจาก Backend จะถูกนำมาแสดงผลในรูปแบบที่เข้าใจง่าย เช่น ตารางกราฟ หรือ แผนภูมิ โดยใช้ไลบรารี เช่น Recharts หรือ Chart.js เพื่อช่วยให้ผู้ใช้สามารถวิเคราะห์ผลคะแนนได้

- การรองรับการทำงานแบบ **Responsive**

ระบบถูกออกแบบให้สามารถใช้งานได้ทั้งบน คอมพิวเตอร์ แท็บเล็ต และโทรศัพท์มือถือ โดยใช้ **Tailwind CSS** เพื่อให้ UI สามารถปรับขนาดได้อัตโนมัติตามอุปกรณ์ของผู้ใช้

- การทำงานร่วมกับ **NGINX**

ในการ Deploy ระบบจริง Frontend ของ ScoreOBE+ จะถูกให้บริการผ่าน **NGINX** ซึ่งทำหน้าที่เป็น **Reverse Proxy** โดยมีข้อดีดังนี้:

- ลดภาระของ Backend โดยแคชไฟล์ Static เช่น HTML, CSS และ JavaScript
- ปรับปรุงประสิทธิภาพและความเร็วในการโหลดหน้าเว็บ
- เพิ่มความปลอดภัยโดยซ่อนโครงสร้างภายในของเซิร์ฟเวอร์จากผู้ใช้ภายนอก

### เทคโนโลยีที่ใช้ใน **Frontend**

- React.js** - ใช้เป็น Core Library สำหรับพัฒนา UI แบบ Component-Based
- Tailwind CSS** - ใช้สำหรับการจัดการสไตร์ UI ที่มีความยืดหยุ่นและง่ายต่อการปรับแต่ง
- React Router** - ใช้จัดการเส้นทาง (Routing) ภายในเว็บแอปพลิเคชัน
- Recharts / Chart.js** - ใช้สร้างกราฟและแผนภูมิแสดงผลข้อมูล
- Axios** - ใช้สำหรับดึงข้อมูลจาก Backend ผ่าน RESTful API
- NGINX** - ใช้เป็น Reverse Proxy เพื่อเพิ่มประสิทธิภาพและความปลอดภัยของระบบ

#### 3.1.3 Backend

Backend ของ ScoreOBE+ ถูกพัฒนาด้วย **NestJS** ซึ่งเป็น Framework บน **Node.js** ที่รองรับ **TypeScript** โดย NestJS ถูกออกแบบมาให้มีโครงสร้างแบบ **Modular** ทำให้การพัฒนาและบำรุงรักษาง่าย อีกทั้งยังรองรับแนวทาง **Dependency Injection** ซึ่งช่วยให้สามารถจัดการการเรียกใช้ Services ต่างๆ ได้

### หน้าที่หลักของ **Backend**

Backend ทำหน้าที่เป็นเซิร์ฟเวอร์กลางที่จัดการการรับส่งข้อมูลระหว่าง Frontend กับแหล่งข้อมูลอื่นๆ โดยมีฟังก์ชันการทำงานหลักดังนี้:

- การรับและประมวลผลคำขอจาก **Frontend**

Backend ให้บริการ **RESTful API** ที่ใช้รับคำขอจากผู้ใช้ผ่าน Frontend โดยมี Middleware เช่น Express ค่อยจัดการ Request และ Response

- การตรวจสอบสิทธิ์ผู้ใช้ (**Authentication and Authorization**)

ระบบใช้ **Microsoft Entra ID** (Azure AD) ในการยืนยันตัวตน และใช้ **JSON Web Token (JWT)** สำหรับจัดการเชสชันและการตรวจสอบสิทธิ์ในระบบ ScoreOBE+

- การจัดการข้อมูลในฐานข้อมูล

Backend เชื่อมต่อกับ **MongoDB** ซึ่งเป็นฐานข้อมูลแบบ NoSQL สำหรับจัดเก็บข้อมูลสำคัญ เช่น คะแนนของนักศึกษา กระบวนการวิชา และรายละเอียดของ TQF3 โดยใช้ **Mongoose** เป็น ORM/ODM สำหรับการจัดการโครงสร้างข้อมูล

- การเชื่อมต่อกับ **CMU MIS API**

Backend มีการเชื่อมต่อกับ **CMU MIS API** เพื่อดึงข้อมูลรายละเอียดของกระบวนการวิชา (เช่น รหัสกระบวนการวิชา ชื่อวิชา และข้อมูลอาจารย์ผู้สอน) ซึ่งนำไปใช้ในการจัดทำเอกสาร **TQF3** ที่ต้องส่งให้มหาวิทยาลัย

- การคำนวณคะแนนและวิเคราะห์ผลการเรียน

Backend รองรับการคำนวณคะแนนของนักศึกษาตาม **Outcome-Based Education (OBE)** โดยสามารถแปลงคะแนนดิบให้เป็นค่าต่าง ๆ เช่น คะแนนเฉลี่ย, ค่ามาตรฐานเบี่ยงเบน, คะแนนเปอร์เซ็นไทล์ และสามารถสร้างสรุปสถิติได้แบบอัตโนมัติ

- การจัดการ **Role-Based Access Control (RBAC)**

Backend มีการกำหนดระดับสิทธิ์ของผู้ใช้ เช่น **Admin, Instructor, Student** เพื่อให้ผู้ใช้แต่ละกลุ่มสามารถเข้าถึงข้อมูลที่เกี่ยวข้องเท่านั้น โดยใช้ Middleware ในการตรวจสอบสิทธิ์ก่อนให้เข้าถึงทรัพยากรของระบบ

### เทคโนโลยีที่ใช้ใน Backend

- NestJS** - Framework สำหรับพัฒนา Backend ที่รองรับ TypeScript
- MongoDB** - ฐานข้อมูลแบบ NoSQL สำหรับจัดเก็บข้อมูลของระบบ
- Mongoose** - ใช้เป็น ODM/ORM สำหรับจัดการโครงสร้างข้อมูล
- Azure AD + JWT** - ใช้สำหรับ Authentication และการจัดการสิทธิ์ผู้ใช้
- CMU MIS API** - ใช้ดึงข้อมูลกระบวนการวิชาจากระบบของมหาวิทยาลัย

#### 3.1.4 Database

ระบบ ScoreOBE+ ใช้ **MongoDB** เป็นฐานข้อมูลหลัก เนื่องจากเป็นฐานข้อมูลแบบ **NoSQL** ที่มีโครงสร้างแบบ **Document-Oriented** ช่วยให้สามารถจัดเก็บข้อมูลที่มีความยืดหยุ่นและรองรับการขยายระบบได้ โดยข้อมูลถูกเก็บในรูปแบบ **JSON-like BSON** ซึ่งช่วยให้การ Query ข้อมูลมีประสิทธิภาพ

### เหตุผลที่เลือกใช้ **MongoDB**

- รองรับโครงสร้างข้อมูลที่ยืดหยุ่น

เนื่องจากข้อมูลผู้ใช้งาน และกระบวนการวิชาอาจมีโครงสร้างที่แตกต่างกันในแต่ละกระบวนการวิชา ซึ่ง MongoDB ช่วยให้สามารถจัดเก็บข้อมูลที่มีรูปแบบต่างกันได้โดยไม่ต้องกำหนด Schema ล่วงหน้า

- รองรับการ **Query** ข้อมูลที่ซับซ้อน

MongoDB มีฟังก์ชัน Aggregation Framework ที่ช่วยให้สามารถประมวลผลข้อมูล เช่น การคำนวณค่าแนวโน้มลี่ของนักศึกษา หรือการสรุปสถิติผลลัพธ์การเรียนรู้

- รองรับการกระจายโหลดและขยายระบบ

MongoDB รองรับการทำ **Sharding** และ **Replication** ซึ่งช่วยให้สามารถกระจายโหลดข้อมูลและเพิ่มประสิทธิภาพการเข้าถึงข้อมูลในระบบที่มีขนาดใหญ่

### 3.1.5 NGINX และ Docker

ระบบใช้งาน **NGINX** เป็น Reverse Proxy เพื่อจัดการทรัพฟิก HTTP และเพิ่มประสิทธิภาพของเซิร์ฟเวอร์ และมีการใช้ **Docker** ในการแยกส่วนการทำงานของแต่ละองค์ประกอบ โดยมี 3 container ได้แก่ Frontend, Backend และ Database

## 3.2 Database Design

### 3.2.1 โครงสร้างของฐานข้อมูลใน MongoDB

ฐานข้อมูลของระบบ ScoreOBE+ ใช้ **MongoDB** เป็นหลักในการจัดเก็บข้อมูล โดยมีการออกแบบคอลเลกชัน (Collections) ต่าง ๆ เพื่อรับฟังก์ชันการทำงานของระบบ ดังนี้

#### AcademicYear Collection

คอลเลกชันนี้จัดเก็บข้อมูลเกี่ยวกับปีการศึกษา

- year** (Number) : ปีการศึกษา เช่น 2023
- semester** (Number) : ภาคการศึกษาที่เกี่ยวข้อง เช่น 1, 2 หรือ 3
- isActive** (Boolean) : สถานะของปีการศึกษาปัจจุบัน

#### Faculties Collection

คอลเลกชันนี้ใช้สำหรับจัดเก็บข้อมูลเกี่ยวกับคณะ

- facultyCode** (String) : รหัสคณะ เช่น "ENG"
- facultyTH** (String) : ชื่อคณะภาษาไทย
- facultyEN** (String) : ชื่อคณะภาษาอังกฤษ
- department** (Object Array) : รายการแผนกวิชาในคณะ
- curriculum** (Object Array) : รายการหลักสูตรของคณะ

## PLO (Program Learning Outcomes) Collection

คอลเลกชันนี้ใช้สำหรับจัดเก็บข้อมูลเกี่ยวกับ PLO

- **name** (String) : ชื่อของ PLO
- **facultyCode** (String) : รหัสคณะที่เกี่ยวข้อง
- **curriculum** (String Array) : รายการหลักสูตรที่ใช้ PLO นี้
- **criteriaTH** (String) : เกณฑ์ภาษาไทย
- **criteriaEN** (String) : เกณฑ์ภาษาอังกฤษ
- **data** (Object Array) : รายการของ PLO Number

## CourseManagement Collection

คอลเลกชันนี้ใช้สำหรับจัดเก็บข้อมูลเกี่ยวกับการจัดการกระบวนวิชา

- **courseNo** (String) : รหัสวิชา
- **courseName** (String) : ชื่อวิชา
- **descTH** (String) : คำอธิบายภาษาไทย
- **descEN** (String) : คำอธิบายภาษาอังกฤษ
- **updatedYear** (Number) : ปีที่อัปเดตล่าสุด
- **updatedSemester** (Number) : ภาคการศึกษาที่อัปเดตล่าสุด
- **type** (Enum) : ประเภทของกระบวนวิชา
- **sections** (Object Array) : รายการของ SectionManagement

## User Collection

คอลเลกชันนี้ใช้สำหรับจัดเก็บข้อมูลของผู้ใช้ระบบ

- **studentId** (String) : รหัสนักศึกษา
- **firstNameTH** (String) : ชื่อจริงภาษาไทย
- **lastNameTH** (String) : นามสกุลภาษาไทย
- **firstNameEN** (String) : ชื่อจริงภาษาอังกฤษ
- **lastNameEN** (String) : นามสกุลภาษาอังกฤษ
- **email** (String) : อีเมลของผู้ใช้
- **facultyCode** (String) : รหัสคณะของผู้ใช้
- **role** (Enum) : บทบาทของผู้ใช้ เช่น Instructor หรือ Student

- **curriculums** (String Array) : รายการหลักสูตรที่เกี่ยวข้องกับผู้ใช้
- **termsOfService** (Boolean) : เงื่อนไขการใช้บริการ
- **enrollCourses** (Object Array) : กระบวนการวิชาที่ลงทะเบียนเรียน

## Course Collection

คอลเลกชันนี้จัดเก็บข้อมูลเกี่ยวกับกระบวนการวิชา

- **year** (Number, required) : ปีการศึกษา
- **semester** (Number, required) : ภาคการศึกษา
- **courseNo** (String, required) : รหัสวิชา
- **courseName** (String, required) : ชื่อวิชา
- **descTH** (String, required) : คำอธิบายภาษาไทย
- **descEN** (String, required) : คำอธิบายภาษาอังกฤษ
- **type** (Enum, required) : ประเภทของกระบวนการวิชา
- **sections** ([Section](#) Array, required) : กลุ่มเรียนของกระบวนการวิชา
- **addFirstTime** (Boolean) : เป็นการเพิ่มกระบวนการวิชานี้เป็นครั้งแรกหรือไม่
- **TQF3** (ObjectId, ref: 'TQF3') : อ้างอิงถึงเอกสาร TQF3
- **TQF5** (ObjectId, ref: 'TQF5') : อ้างอิงถึงเอกสาร TQF5

## TQF3 Collection

คอลเลกชันนี้จัดเก็บเอกสาร TQF3 ซึ่งประกอบด้วยหลายส่วน

- **status** (Enum, required) : สถานะของเอกสาร
- **part1** (Object) : ส่วนที่ 1 ของ TQF3
- **part2** (Object) : ส่วนที่ 2 ของ TQF3
- **part3** (Object) : ส่วนที่ 3 ของ TQF3
- **part4** (Object) : ส่วนที่ 4 ของ TQF3
- **part5** (Object) : ส่วนที่ 5 ของ TQF3
- **part6** (Object) : ส่วนที่ 6 ของ TQF3
- **part7** (Object) : ส่วนที่ 7 ของ TQF3

## TQF5 Collection

คอลเลกชันนี้จัดเก็บเอกสาร TQF5 ซึ่งประกอบด้วยหลายส่วน

- **status** (Enum, required) : สถานะของเอกสาร
- **method** (Enum) : วิธีการที่ใช้
- **assignmentsMap** (Object Array) : การจับคู่ระหว่างการประเมินและงานที่มีอุบหมาย
  - **eval** (String) : ชื่อการประเมิน
  - **assignment** (String Array) : ชื่องานที่มีอุบหมาย
- **part1** (Object) : ส่วนที่ 1 ของ TQF5
- **part2** (Object) : ส่วนที่ 2 ของ TQF5
- **part3** (Object) : ส่วนที่ 3 ของ TQF5

### 3.2.2 โครงสร้างออบเจกต์ในระบบฐานข้อมูล

ในระบบฐานข้อมูล **MongoDB** ที่ใช้ในโปรเจกต์นี้ ประกอบด้วยคอลเลกชันหลายคอลเลกชัน โดยแต่ละคอลเลกชันจะมีออบเจกต์ที่มีโครงสร้างแตกต่างกันไป ออบเจกต์เหล่านี้ถูกกำหนดโดยใช้ Schema ในภาษา TypeScript ผ่าน Mongoose ซึ่งเป็นไลบรารีสำหรับการจัดการฐานข้อมูล MongoDB

ออบเจกต์เหล่านี้มีความสัมพันธ์กันในลักษณะต่างๆ เช่น การอ้างอิงถึงกัน (Reference) หรือการผังข้อมูลเข้าไว้ด้วยกัน (Embedding) ซึ่งเป็นรูปแบบการออบแบบฐานข้อมูลที่พิเศษอยู่ใน MongoDB ต่อไปนี้เป็นรายละเอียดของออบเจกต์ต่างๆ ที่ใช้ในระบบ

#### Score Object

ออบเจกต์นี้ใช้สำหรับจัดเก็บข้อมูลเกี่ยวกับคะแนนของงานที่มีอุบหมาย โดยจะถูกใช้เป็นส่วนหนึ่งของข้อมูลนักศึกษาในคอลเลกชัน Section

- **assignmentName** (String) : ชื่อของงานที่มีอุบหมาย
- **questions** (Object Array) : รายการคำถามพร้อมคะแนน
  - **name** (String) : ชื่อของคำถาม
  - **score** (Number) : คะแนนที่ได้รับสำหรับคำถาม

#### Question Object

ออบเจกต์นี้กำหนดโครงสร้างของคำถาม ซึ่งจะถูกใช้เป็นส่วนหนึ่งของ Assignment Object

- **name** (String, required) : ชื่อของคำถาม
- **desc** (String) : คำอธิบายของคำถาม
- **fullScore** (Number, required, min: 0) : คะแนนเต็มที่เป็นไปได้

## Assignment Object

ออบเจกต์นี้กำหนดงานที่มีขอบหมายพร้อมคำถาม ซึ่งจะถูกใช้เป็นส่วนหนึ่งของคอลเลกชัน Section

- **name** (String, required) : ชื่อของงานที่มีขอบหมาย
- **isPublish** (Boolean, default: false) : สถานะการประกาศ
- **weight** (Number, required, default: 1) : น้ำหนักของงานที่มีขอบหมาย
- **questions** (Question Array, required) : รายการคำถามในงานที่มีขอบหมาย

## Section Object

ออบเจกต์นี้จัดเก็บข้อมูลเกี่ยวกับกลุ่มเรียน ซึ่งเป็นส่วนหนึ่งของคอลเลกชัน Course

- **sectionNo** (Number, required) : หมายเลขกลุ่มเรียน
- **curriculum** (String) : หลักสูตรที่กลุ่มเรียนนี้สังกัด
- **addFirstTime** (Boolean) : เป็นการเพิ่มกลุ่มเรียนนี้เป็นครั้งแรกหรือไม่
- **isActive** (Boolean, required, default: true) : สถานะการใช้งาน
- **instructor** (ObjectId, ref: 'User', required) : อาจารย์ผู้สอนหลัก
- **coInstructors** (ObjectId Array, ref: 'User') : อาจารย์ผู้สอนร่วม
- **students** (Array) : นักศึกษาที่ลงทะเบียนในกลุ่มเรียน
  - **student** (ObjectId, ref: 'User') : อ้างอิงถึงนักศึกษา
  - **scores** (Score Array) : คะแนนของนักศึกษา
- **assignments** (Assignment Array) : งานที่มีขอบหมายสำหรับกลุ่มเรียน
- **topic** (String) : หัวข้อของกลุ่มเรียน
- **TQF3** (ObjectId, ref: 'TQF3') : อ้างอิงถึงเอกสาร TQF3
- **TQF5** (ObjectId, ref: 'TQF5') : อ้างอิงถึงเอกสาร TQF5

### 3.2.3 ตัวอย่างโครงสร้างข้อมูล

ตัวอย่างเอกสาร (Document) ที่ใช้เก็บข้อมูลผู้ใช้งานใน Collection Users โดยในตัวอย่างเป็น role Student

```
{  
  "_id": {  
    "$oid": "665e147dcee0f327c0c00b50"  
  },
```

```

"studentId": "640610672",
"firstNameTH": "ສວີ່ພ",
"lastNameTH": "ຈາກີກພູນພລ",
"firstNameEN": "Sawit",
"lastNameEN": "Charuekpoonpol",
"email": "sawit_cha@cmu.ac.th",
"facultyCode": "06",
"role": "Admin",
"enrollCourses": [
{
    "year": 2567,
    "semester": 2,
    "courses": [
        {
            "course": {
                "$oid": "67322fb6f67986b40e1bc948"
            },
            "section": 1
        },
        {
            "course": {
                "$oid": "67838142432717f4e5032d70"
            },
            "section": 640
        },
        {
            "course": {
                "$oid": "67d53c9444027b2c162e6102"
            },
            "section": 1
        }
    ]
},
],
"curriculums": [],
"termsOfService": true
}

```

### 3.3 การคำนวณคะแนน CLO (Course Learning Outcomes)

การคำนวณคะแนน CLO เป็นกระบวนการสำคัญในการประเมินผลการเรียนรู้ของนักศึกษาตามผลลัพธ์การเรียนรู้ที่กำหนดไว้ในกระบวนการวิชา โดยมีขั้นตอนการดำเนินการที่เป็นระบบดังนี้

#### 3.3.1 ขั้นตอนการเตรียมการก่อนการคำนวณ

##### 1. การทำ Assessment Mapping (TQF3 Part 4)

- เป็นขั้นตอนแรกและสำคัญที่สุดในการคำนวณคะแนน CLO
- ในขั้นตอนนี้ อาจารย์ผู้สอนจะต้องกำหนดความสัมพันธ์ระหว่าง CLO แต่ละข้อกับวิธีการประเมิน (Assessment) ต่างๆ
- สำหรับแต่ละ CLO จะต้องระบุว่าจะใช้ Assessment ใดบ้างในการประเมิน และแต่ละ Assessment มีน้ำหนัก (Weight) เท่าใดในการคิดคะแนน CLO นั้นๆ
- ผลลัพธ์ของขั้นตอนนี้คือตารางความสัมพันธ์ที่แสดงว่า CLO แต่ละข้อจะถูกประเมินด้วย Assessment ใดบ้าง และแต่ละ Assessment มีน้ำหนักเท่าใด

CLO Description / Evaluation Topic	assessment midterm	assessment final	Total CLO (%)
CLO-1 CLO1	0%	20%	20%
CLO-2 CLO2	15%	0%	15%
CLO-3 CLO3	35%	5%	40%
CLO-4 CLO4	0%	25%	25%
Total Assessment (%)	50 / 50%	50 / 50%	100%

รูปที่ 3.2: Assessment Mapping

##### 2. การทำ Question Mapping (TQF5 Part 2)

- หลังจากกำหนด Assessment Mapping แล้ว ขั้นตอนต่อไปคือการระบุว่าคำถามใดใน Assessment แต่ละข้อจะใช้ในการประเมิน CLO ข้อใด
- ในขั้นตอนนี้ อาจารย์ผู้สอนจะต้องเชื่อมโยงคำถาม (Question) แต่ละข้อใน Assessment เข้ากับ CLO ที่เกี่ยวข้อง
- การทำ Question Mapping นี้มีความสำคัญเนื่องจากใน Assessment หนึ่งๆ อาจมีคำถามที่ประเมิน CLO หลายข้อ

- ผลลัพธ์ของขั้นตอนนี้คือรายการคำถ้าในแต่ละ Assessment ที่ใช้ประเมิน CLO แต่ละข้อ

The screenshot shows the ScoreOBE platform interface for mapping assessment questions to Course Learning Outcomes (CLOs). The left sidebar shows the course navigation: Back to Your Course, 261999 (2/67), Course CPE Mock, Evaluations, Sections, Roster, TQF 3, and TQF 5 (selected). The main content area is titled 'TQF 5' and 'Part 2 - การเลือกหัวข้อประเมินตามถูกประสงค์ของระบบวิชา' (Assessment tool mapping to CLO). It lists CLO 1, CLO 2, CLO 3, and CLO 4. For CLO 1, there is a 'Final' section with '1 - design' and '2 - testing' checked. For CLO 2, there is a 'Midterm' section with '1 - Security', '2 - Page Replacement', '3 - Paging', and '4 - Protection' checked. A sidebar on the right shows 'CLO 1', 'CLO 2', 'CLO 3', and 'CLO 4'. At the bottom right, there are 'Saved' and 'Save Part 2' buttons.

รูปที่ 3.3: Map Question in Assessment with Course CLOs

### 3. การกำหนดช่วงคะแนน (Set Range Assessment Tool)

- การกำหนดช่วงคะแนน: อาจารย์ผู้สอนจะต้องกำหนดช่วงคะแนนดิบในช่วงใดจะถูกแปลงเป็นคะแนนมาตรฐาน 0-4 โดยทั่วไป ระบบจะกำหนดช่วงคะแนนเริ่มต้นให้เป็นช่วงที่เท่าๆ กัน แต่อาจารย์ผู้สอนสามารถปรับเปลี่ยนได้ตามความเหมาะสม
- การเปรียบเทียบผลการประเมินจาก Assessment ที่คะแนนเต็มต่างกัน: การกำหนดช่วงคะแนนนี้จะช่วยให้สามารถเปรียบเทียบผลการประเมินจาก Assessment ที่มีคะแนนเต็มต่างกันได้ เช่น คะแนนเต็ม 9 สำหรับ Midterm หรือคะแนนเต็ม 2 สำหรับ Homework
- การใช้เส้นจำนวน (Number Line): ในการกำหนดช่วงคะแนนบนเส้นจำนวน จะมีการใช้ **closed interval** (การรวมจุดที่ทางซ้าย) และ **open interval** (การไม่รวมจุดที่ทางขวา) ตัวอย่างเช่น:

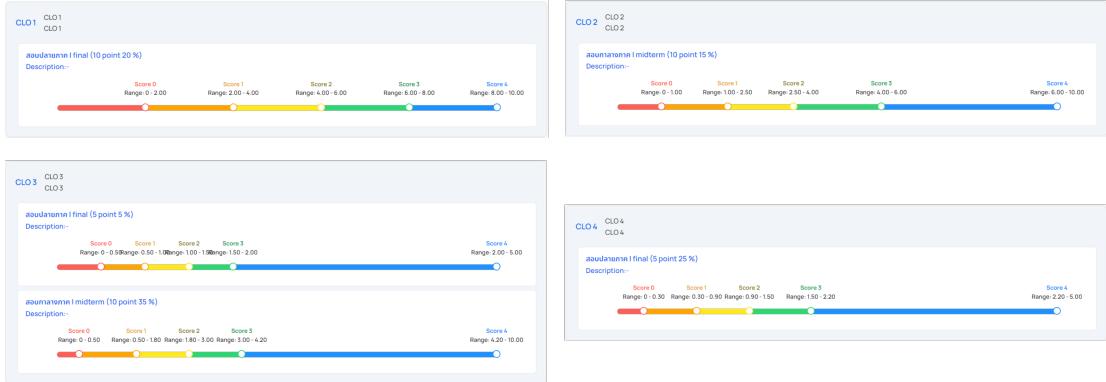
$$[0, 1] \cup (1, 2] \cup (2, 3] \cup (3, 6] \cup (6, 9]$$

ซึ่งหมายความว่า:

- ช่วง  $[0, 1]$  เป็น **closed interval** ที่รวม 0 และไม่รวม 1
- ช่วง  $[1, 2)$  เป็น **closed interval** ที่รวม 1 และไม่รวม 2
- ช่วง  $[2, 3)$  เป็น **closed interval** ที่รวม 2 และไม่รวม 3
- ช่วง  $[3, 6)$  เป็น **closed interval** ที่รวม 3 และไม่รวม 6
- ช่วง  $[6, 9]$  เป็น **closed interval** ที่รวมทั้ง 6 และ 9

การใช้สีสันจำนวนเช่นนี้ทำให้การกำหนดช่วงคะแนนมีความชัดเจนและสามารถแปลงคะแนนดิบให้เป็นคะแนนมาตรฐานได้อย่างแม่นยำ

- ผลลัพธ์ของการกำหนดช่วงคะแนน: ผลลัพธ์จากการกำหนดช่วงคะแนนคือการสร้างตารางที่แสดงช่วงคะแนนดิบที่สอดคล้องกับคะแนนมาตรฐาน 0-4 สำหรับแต่ละ Assessment Tool ซึ่งจะช่วยให้อาจารย์สามารถแปลงคะแนนดิบให้เป็นคะแนนมาตรฐานได้อย่างถูกต้อง



รูปที่ 3.4: Set Range Assessment Tool

### 3.3.2 กระบวนการคำนวณคะแนน CLO

เมื่อเตรียมการครบถ้วน 3 ขั้นตอนข้างต้นแล้ว สามารถคำนวณคะแนน CLO ได้ตามขั้นตอนดังนี้:

#### 1. การแปลงคะแนนดิบเป็นคะแนนมาตรฐาน

- นำคะแนนดิบของนักศึกษาแต่ละคนในแต่ละ Question ที่เกี่ยวข้องกับ CLO ที่ต้องการคำนวณมาพิจารณา
- รวมคะแนนดิบของ Question ที่อยู่ใน Assessment เดียวกันเข้าด้วยกัน
- นำคะแนนรวมที่ได้มาเทียบกับช่วงคะแนนที่กำหนดไว้ เพื่อแปลงเป็นคะแนนมาตรฐาน 0-4

#### 2. การคำนวณคะแนน CLO รายบุคคล

- สำหรับนักศึกษาแต่ละคน คำนวณคะแนน CLO โดยใช้สูตร:

$$\text{คะแนน CLO} = \frac{\sum (\text{คะแนนมาตรฐานของ Assessment} \times \text{น้ำหนัก})}{\sum \text{น้ำหนัก}} \quad (3.1)$$

- คะแนนที่ได้จะอยู่ในช่วง 0-4 โดยหากคำนวณออกมาแล้วมีค่าทศนิยมจะทำการปัดลงทั้งหมด ซึ่งส่วนท่อนหลังจะดับความสามารถของนักศึกษาในการบรรลุ CLO ที่กำหนด โดยการประเมินระดับความสามารถจะเป็นดังนี้: 0 หมายถึง แย่มาก, 1 หมายถึง แย่, 2 หมายถึง ปานกลาง, 3 หมายถึง ดี, และ 4 หมายถึง ดีมาก

### 3. การคำนวณคะแนน CLO เฉลี่ยของกระบวนวิชา

- เมื่อได้คะแนน CLO ของนักศึกษาทุกคนแล้ว สามารถคำนวณคะแนน CLO เฉลี่ยของกระบวนวิชา ได้โดยใช้สูตร:

$$\text{คะแนน CLO เฉลี่ย} = \frac{\sum \text{คะแนน CLO ที่ปัดเศษนิยมของนักศึกษาทุกคน}}{\text{จำนวนนักศึกษา}} \quad (3.2)$$

- คะแนน CLO เฉลี่ยนี้จะใช้ในการประเมินประสิทธิภาพของการจัดการเรียนการสอนและนำไปสู่การปรับปรุงในอนาคต

#### 3.3.3 การคำนวณคะแนน CLO (Course Learning Outcomes)

##### ตัวอย่างการคำนวณคะแนน CLO

การคำนวณคะแนน CLO เป็นกระบวนการทำงานสำคัญในการประเมินผลการเรียนรู้ของนักศึกษา เพื่อให้เข้าใจได้ง่าย เราจะแสดงตัวอย่างการคำนวณตามขั้นตอนต่างๆ ดังนี้

##### 1. ข้อมูลพื้นฐาน

- กระบวนวิชา: XXX101
- CLO ที่ต้องการประเมิน: CLO 1
- จำนวนนักศึกษา: 3 คน (640610xx1, 640610xx2, 640610xx3)
- วิธีการประเมินและน้ำหนัก:
  - Midterm (น้ำหนัก 5%) - ใช้ข้อสอบกลางภาคข้อที่ 1 และ 2
  - Final (น้ำหนัก 10%) - ใช้ข้อสอบปลายภาคข้อที่ 1
  - Homework (น้ำหนัก 5%) - ใช้แบบฝึกหัดข้อที่ 2

2. คะแนนดิบของนักศึกษา ตารางด้านล่างแสดงคะแนนดิบของนักศึกษาทั้ง 3 คน โดยเฉลล์ที่มีการระบายน้ำหนักคะแนนที่เกี่ยวข้องกับการประเมิน CLO 1

รหัสนักศึกษา	Midterm (15 คะแนน)			Final (10 คะแนน)	Homework (6 คะแนน)	
	ข้อ 1 (5)	ข้อ 2 (4)	ข้อ 3 (6)		ข้อ 1 (4)	ข้อ 2 (2)
640610xx1	3	4	5	9	2	0.75
640610xx2	2	1	3	7	1	0.65
640610xx3	5	2	4	8	3	0.5

ตารางที่ 3.1: คะแนนดิบของนักศึกษา

3. การรวมคะแนนตามวิธีการประเมิน เราจะรวมคะแนนเฉพาะข้อที่เกี่ยวข้องกับ CLO 1 ในแต่ละวิธีการประเมิน:

รหัสนักศึกษา	Midterm	Final	Homework
	(ข้อ 1+2 = 9 คะแนน)	(ข้อ 1 = 10 คะแนน)	(ข้อ 2 = 2 คะแนน)
640610xx1	$3 + 4 = 7$	9	0.75
640610xx2	$2 + 1 = 3$	7	0.65
640610xx3	$5 + 2 = 7$	8	0.5

ตารางที่ 3.2: คะแนนรวมตามวิธีการประเมิน

4. เกณฑ์การแปลงคะแนนเป็นมาตรฐาน 0-4 เพื่อให้สามารถเปรียบเทียบคะแนนจากวิธีการประเมินที่มีคะแนนเต็มต่างกันได้ เราจะแปลงคะแนนดิบให้เป็นคะแนนมาตรฐาน 0-4 ตามเกณฑ์ดังนี้:

วิธีการประเมิน	Score 0	Score 1	Score 2	Score 3	Score 4
Midterm (9 คะแนน)	0 - 1	1 - 2	2 - 3	3 - 6	6 - 9
Final (10 คะแนน)	0 - 3	3 - 6	6 - 8	8 - 9	9 - 10
Homework (2 คะแนน)	0 - 0.2	0.2 - 0.4	0.4 - 0.6	0.6 - 0.8	0.8 - 2

ตารางที่ 3.3: เกณฑ์การแปลงคะแนนเป็นมาตรฐาน 0-4

5. ผลการแปลงคะแนนเป็นมาตรฐาน 0-4 เมื่อนำคะแนนรวมมาเทียบกับเกณฑ์การแปลงคะแนน จะได้คะแนนมาตรฐานดังนี้:

รหัสนักศึกษา	Midterm (5%)	Final (10%)	Homework (5%)
640610xx1	4	4	3
640610xx2	3	2	3
640610xx3	4	3	2

ตารางที่ 3.4: คะแนนมาตรฐาน 0-4 ของแต่ละวิธีการประเมิน

## 6. การคำนวณคะแนน CLO รายบุคคล

$$\text{คะแนน CLO} = \frac{\sum (\text{คะแนนมาตรฐาน} \times \text{น้ำหนัก})}{\sum \text{น้ำหนัก}} \quad (3.3)$$

รหัสนักศึกษา	การคำนวณ	คะแนน CLO 1
640610xx1	$\frac{(4 \times 5) + (4 \times 10) + (3 \times 5)}{5+10+5} = \frac{20+40+15}{20} = \frac{75}{20}$	3.75
640610xx2	$\frac{(3 \times 5) + (2 \times 10) + (3 \times 5)}{5+10+5} = \frac{15+20+15}{20} = \frac{50}{20}$	2.50
640610xx3	$\frac{(4 \times 5) + (3 \times 10) + (2 \times 5)}{5+10+5} = \frac{20+30+10}{20} = \frac{60}{20}$	3.00

ตารางที่ 3.5: การคำนวณคะแนน CLO 1 รายบุคคล

## 7. การคำนวณคะแนน CLO เฉลี่ย คะแนน CLO เฉลี่ยของระบบวิชาคำนวณจากค่าเฉลี่ยของคะแนน CLO ของนักศึกษาทุกคน:

$$\text{คะแนน CLO 1 เฉลี่ย} = \frac{3.75 + 2.50 + 3.00}{3} = \frac{9.25}{3} \approx 3.08$$

## 8. สรุปผลการประเมิน CLO 1 จากการคำนวณข้างต้น สรุปได้ว่า:

- นักศึกษารหัส 640610xx1 มีคะแนน CLO 1 = 3.75 ≈ 3 (ดี)
- นักศึกษารหัส 640610xx2 มีคะแนน CLO 1 = 2.50 ≈ 2 (ปานกลาง)
- นักศึกษารหัส 640610xx3 มีคะแนน CLO 1 = 3.00 ≈ 3 (ดี)
- คะแนน CLO 1 เฉลี่ยของระบบวิชา = 2.67

คะแนนเฉลี่ยนี้จะถูกนำไปใช้ในการประเมินประสิทธิภาพของการจัดการเรียนการสอนและการบรรยาย ลักษณะเรียนรู้ที่กำหนดไว้ในระบบวิชา

### 3.3.4 การคำนวณคะแนน PLO (Program Learning Outcomes)

หลังจากคำนวณคะแนน CLO ของทุกข้อในกระบวนการวิชาเรียบร้อยแล้ว ขั้นตอนต่อไปคือการคำนวณคะแนน PLO ซึ่งเป็นผลลัพธ์การเรียนรู้ในระดับหลักสูตร

### 3.4 การคำนวณคะแนน PLO

เมื่อคำนวณคะแนน CLO เสร็จสิ้นในแต่ละข้อ ระบบจะนำคะแนนเหล่านั้นมาคำนวณหาคะแนนของ PLO ต่อไป โดยคะแนน PLO แต่ละข้อจะมาจากการคำนวณของคะแนน CLO ที่เขื่อมโยงกับ PLO ข้อนั้น

#### 3.4.1 ตัวอย่างการคำนวณ

CLO Description (4 CLOs)	PLO-1	PLO-2*	PLO-3*	PLO-4*	PLO-5	PLO-6*	PLO-7*
CLO-1 CLO 1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CLO-2 CLO 2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CLO-3 CLO 3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CLO-4 CLO 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

รูปที่ 3.5: Curriculum Mapping

สมมติว่า ในวิชาหนึ่ง มีการเขื่อมโยง PLO กับ CLO ดังรูปที่ 3.5 และคะแนน CLO ในแต่ละข้อเป็นดังนี้

CLO 1	CLO 2	CLO 3	CLO 4
1.83	2.77	3.04	2.86

ตารางที่ 3.6: คะแนน CLO

$$\text{คะแนน PLO} = \frac{\sum \text{คะแนน CLO ที่เขื่อมโยงกับ PLO นั้น}}{\text{จำนวน CLO ที่เขื่อมโยงกับ PLO นั้น}} \quad (3.4)$$

จะสามารถคำนวณคะแนน PLO ในแต่ละข้อได้ดังนี้

- **PLO 1:** ไม่มีการเชื่อมโยงกับ CLO จึงไม่สามารถคำนวณคะแนนได้
- **PLO 2:**  $\frac{1.83+3.04}{2} = \frac{4.87}{2} = 2.44$  คะแนน
- **PLO 3:**  $\frac{2.77}{1} = 2.77$  คะแนน
- **PLO 4:**  $\frac{1.83}{1} = 1.83$  คะแนน
- **PLO 5:** ไม่มีการเชื่อมโยงกับ CLO จึงไม่สามารถคำนวณคะแนนได้
- **PLO 6:**  $\frac{2.86}{1} = 2.86$  คะแนน
- **PLO 7:**  $\frac{2.77}{1} = 2.77$  คะแนน

PLO	คะแนน	หมายเหตุ
PLO 1	-	ไม่มีการเชื่อมโยงกับ CLO
PLO 2	2.44	เชื่อมโยงกับ CLO 1, CLO 3
PLO 3	2.77	เชื่อมโยงกับ CLO 2
PLO 4	1.83	เชื่อมโยงกับ CLO 1
PLO 5	-	ไม่มีการเชื่อมโยงกับ CLO
PLO 6	2.86	เชื่อมโยงกับ CLO 4
PLO 7	2.77	เชื่อมโยงกับ CLO 2

ตารางที่ 3.7: สรุปคะแนน PLO

### 3.5 API Endpoint

เอกสาร API สามารถติดต่อเพื่อขอดูเพิ่มเติมได้ที่ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

## บทที่ 4

### การทดลองและผลลัพธ์

#### 4.1 การทดสอบความถูกต้องของข้อมูลหลักสูตรและกระบวนวิชา

เนื่องจากหลักสูตรและกระบวนวิชามีความสำคัญต่อการดำเนินงานของ ScoreOBE+ ในการประเมินผลคุณภาพแบบแผน วิเคราะห์ผลสัมฤทธิ์ของนักศึกษา และเชื่อมโยงกับกรอบมาตรฐานคุณวุฒิระดับอุดมศึกษา (TQF) ข้อมูลดังกล่าวที่ใช้ใน ScoreOBE+ ได้รับการนำเข้าผ่าน API ซึ่งพัฒนาโดยฝ่ายพัฒนาระบบทุกโน้ตบุ๊ก สำนักบริการเทคโนโลยีสารสนเทศ มหาวิทยาลัยเชียงใหม่ โดยข้อมูลทั้งหมดมาจาก CMU MIS ระบบสารสนเทศกลางของมหาวิทยาลัยที่ใช้จัดเก็บและบริหารข้อมูลด้านวิชาการ รวมถึงโครงสร้างหลักสูตร กระบวนวิชา และข้อมูลที่เกี่ยวข้องกับการเรียนการสอน เพื่อให้มั่นใจได้ว่าโครงสร้างและเนื้อหาของข้อมูลสอดคล้องกันทั้งหมด ทั้งในส่วนของรหัสวิชา ชื่อกระบวนวิชา หน่วยกิต และข้อกำหนดของหลักสูตร

#### 4.2 การทดสอบการนำเข้าค่าแบบแผน

ข้อมูลแบบแผนของนักศึกษามีความสำคัญมากต่อการประเมินผลสัมฤทธิ์ทางการเรียน ระบบ Score-OBE+ รองรับการอัปโหลดไฟล์ค่าแบบแผนในรูปแบบ CSV และ XLSX เพื่อให้สามารถนำเข้าข้อมูลได้ โดยได้ดำเนินการทดสอบในหัวข้อต่อไปนี้

##### 4.2.1 การนำเข้าไฟล์ CSV/XLSX ตามรูปแบบที่กำหนด

ระบบต้องรองรับการนำเข้าไฟล์ค่าแบบแผนที่มีโครงสร้างมาตรฐาน เช่น รหัสนักศึกษา คะแนนดิบ คะแนน เทิร์น และเกณฑ์การให้คะแนน การทดสอบได้ดำเนินการโดยใช้ไฟล์ตัวอย่างที่มีข้อมูลที่ถูกต้องและไฟล์ที่มีข้อผิดพลาด เช่น ค่าที่หายไปหรือข้อมูลที่ไม่เป็นตัวเลข เพื่อตรวจสอบว่าระบบสามารถแจ้งเตือนข้อผิดพลาดได้อย่างถูกต้อง

##### 4.2.2 การตรวจสอบความถูกต้องของข้อมูลที่นำเข้า

ระบบต้องสามารถตรวจสอบว่าข้อมูลที่นำเข้ามีความครบถ้วนและถูกต้อง เช่น รหัสนักศึกษาถูกต้องตามรูปแบบ คะแนนอยู่ในช่วงที่กำหนด และไม่มีค่าที่ขาดหาย การทดสอบนี้รวมถึงการแจ้งเตือนข้อผิดพลาดให้ผู้ใช้สามารถแก้ไขข้อมูลก่อนดำเนินการต่อ

#### 4.3 การประเมินผลและวิเคราะห์ค่าแบบแผน

หลังจากดำเนินการนำเข้าค่าแบบแผนเรียบร้อยแล้ว ระบบต้องสามารถประเมินผลและวิเคราะห์ข้อมูลได้อย่างแม่นยำและเป็นระบบ เพื่อให้ผู้สอนสามารถประเมินผลสัมฤทธิ์ทางการเรียนของนักศึกษาได้อย่างถูกต้อง

#### 4.3.1 การคำนวณสถิติอัตโนมัติ

ระบบได้รับการออกแบบให้สามารถคำนวณค่าสถิติพื้นฐาน เช่น ค่าเฉลี่ย (Mean), ค่ามัธยฐาน (Median) และค่าความเบี่ยงเบนมาตรฐาน (Standard Deviation) ได้โดยอัตโนมัติ เพื่อช่วยให้ผู้สอนสามารถวิเคราะห์ผลการเรียนของนักศึกษาได้สะดวกยิ่งขึ้น การทดสอบความถูกต้องของการคำนวณดำเนินการโดยใช้สูตรใน Microsoft Excel เพื่อเปรียบเทียบผลลัพธ์และยืนยันความแม่นยำของค่าทางสถิติที่ระบบคำนวณ

#### 4.3.2 การวิเคราะห์การจับคู่คะแนนกับ TQF 3 และ TQF 5

เพื่อให้สามารถประเมินผลการเรียนรู้ของนักศึกษาได้อีก ระบบต้องสามารถเข้ามายังคะแนนของนักศึกษา กับกรอบมาตรฐานผลลัพธ์การเรียนรู้ตาม TQF 3 และ TQF 5 ได้อย่างถูกต้อง การทดสอบในส่วนนี้ดำเนินการโดยใช้ Microsoft Excel ในการคำนวณค่าต่าง ๆ เพื่อตรวจสอบว่าค่าที่ระบบคำนวณสามารถสะท้อนถึงผลลัพธ์ของการเรียนของนักศึกษา

#### 4.3.3 การแสดงผลคะแนนของนักศึกษาในรูปแบบการกระจายคะแนน

ScoreOBE+ ใช้ source code จาก Grader powered by Andrew C. Myers (Cornell University)[18] ในการแสดงผลการแจกแจงคะแนน เพื่อให้มั่นใจได้ว่ากราฟการกระจายตัวคะแนนแสดงผลในรูปแบบที่ถูกต้อง

### 4.4 การทดสอบการใช้งานจากผู้ใช้จริง

เพื่อประเมินประสิทธิภาพของ ScoreOBE+ และวัดระดับความพึงพอใจของผู้ใช้ ระบบได้ดำเนินการทดสอบการใช้งานจริงกับกลุ่มนักศึกษาและอาจารย์ โดยให้ผู้เข้าร่วมทดสอบใช้งานระบบและแสดงความคิดเห็นผ่านแบบสอบถามออนไลน์ การทดสอบนี้มุ่งเน้นไปที่ประสบการณ์การใช้งาน ความสามารถในการเข้าถึงข้อมูล ความเข้าใจผลการเรียนรู้ และประโยชน์ที่ได้รับจาก ScoreOBE+

#### 4.4.1 วัตถุประสงค์ของการทดสอบ

การทดสอบการใช้งานจากผู้ใช้จริงมีวัตถุประสงค์หลักดังต่อไปนี้

- ประเมินความสะดวกในการใช้งานของ ScoreOBE+ สำหรับทั้งนักศึกษาและอาจารย์
- ตรวจสอบว่าอินเทอร์เฟซของระบบมีความเข้าใจง่าย และสามารถเข้าถึงฟังก์ชันต่าง ๆ ได้โดยไม่ต้องมีการแนะนำเพิ่มเติม
- วิเคราะห์ประสบการณ์ของผู้ใช้เกี่ยวกับการแสดงผลคะแนน สถิติ และกราฟการแจกแจงคะแนน
- ประเมินว่าระบบสามารถช่วยให้นักศึกษาเข้าใจผลการเรียนรู้และพัฒนาการของตนเองได้หรือไม่
- ตรวจสอบว่าระบบช่วยอำนวยความสะดวกให้กับอาจารย์ในการบริหารจัดการคะแนนและวิเคราะห์ผลลัพธ์ของนักศึกษาได้อย่างมีประสิทธิภาพหรือไม่
- รวบรวมข้อเสนอแนะและความคิดเห็นเพื่อใช้ในการปรับปรุงระบบให้ดียิ่งขึ้น

#### 4.4.2 กลุ่มตัวอย่างและวิธีการทดสอบ

##### กลุ่มอาจารย์

กลุ่มตัวอย่างที่เข้าร่วมการทดสอบประกอบด้วยคณาจารย์จากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ซึ่งมีบทบาทสำคัญในการใช้ ScoreOBE+ ในการบริหารจัดการและวิเคราะห์ผลการเรียนรู้ของนักศึกษา โดยอาจารย์ที่เข้าร่วมการทดสอบในครั้งนี้ ได้แก่

- ผู้ช่วยศาสตราจารย์ โอม โพธิกานนท์ อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์
- ผู้ช่วยศาสตราจารย์ ดร.นวดนัย คุณเลิศกิจ อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์
- ผู้ช่วยศาสตราจารย์ ดร.ธนาทิพย์ จันทร์คง อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์
- ผู้ช่วยศาสตราจารย์ ดร.ยุทธพงษ์ สมจิต อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์
- รองศาสตราจารย์ ดร.ศันสนีย์ เอื้อพันธ์วิริยะกุล อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์
- ผู้ช่วยศาสตราจารย์ ดร.นิรันดร์ พิสุทธอานันท์ อาจารย์ประจำภาควิชาวิศวกรรมอุตสาหการ

ในการทดสอบระบบกับกลุ่มอาจารย์ ผู้เข้าร่วมได้ทำการใช้งาน ScoreOBE+ ผ่านอุปกรณ์คอมพิวเตอร์และแล็ปท็อป โดยมีการประเมินประสิทธิภาพและความสอดคล้องในการเข้าถึงฟังก์ชันหลักของระบบ ซึ่งรวมถึง

- การสร้างกระบวนการวิชาและนำเข้ารายชื่อนักศึกษา
- การนำเข้าคะแนนและประกาศคะแนน
- การนำออกคะแนนในรูปแบบต่าง ๆ
- การดูคะแนนของนักศึกษาและการวิเคราะห์ค่าทางสถิติ
- การตรวจสอบการกระจายคะแนนและแสดงผลในรูปแบบอิสโทแกรม
- การเข้มโถง **Course Learning Outcomes (CLO)** กับ **Assessment Tools**
- การเข้มโถง **CLO** กับ **Program Learning Outcomes (PLO)**
- การจัดทำเอกสาร **TQF3** และ **TQF5**
- การนำเข้าเกรดของนักศึกษาและการปรับช่วงคะแนนของ Assessment Tools ตามแต่ละ **CLO**
- การนำออกเอกสาร **TQF3, TQF5** และข้อมูลสำหรับจัดทำ **TQF7**
- การจัดการกระบวนการวิชาในแต่ละหลักสูตร
- การจัดการชุด **Program Learning Outcomes (PLO)** ในแต่ละหลักสูตร
- การจัดการเข้มโถง **CLO** กับ **Program Learning Outcomes (PLO) Require** ของทุกกระบวนการวิชา ในแต่ละหลักสูตร
- การจัดการเทอมและปีการศึกษา
- การจัดการผู้ดูแลหลักสูตร และผู้ดูแลระบบ

- การจัดการหลักสูตร

หลังจากการทดลองใช้งาน อาจารย์ได้ให้ข้อเสนอแนะผ่านผู้พัฒนาโดยตรง เพื่อปรับปรุงและเพิ่มประสิทธิภาพของระบบให้ดียิ่งขึ้น ทั้งในด้านประสบการณ์การใช้งานและการตอบสนองต่อความต้องการที่แท้จริงของผู้สอน

### กลุ่มนักศึกษา

กลุ่มตัวอย่างที่เข้าร่วมการทดลองประกอบด้วยนักศึกษาที่ลงทะเบียนเรียนในกระบวนวิชาต่าง ๆ ภายในมหาวิทยาลัยเชียงใหม่ เทอมการศึกษา 2/2567 จำนวน 803 คน (ข้อมูล ณ วันที่ 21 มีนาคม พ.ศ. 2568) โดยมีกลุ่มตัวอย่างจากนักศึกษาที่เรียนในกระบวนวิชาดังต่อไปนี้

- นักศึกษาจากหลายคณะ ที่ลงทะเบียนเรียนในกระบวนวิชา **261111: Internet and Online Community in the Age of AI**
- นักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ที่ลงทะเบียนเรียนในกระบวนวิชา **261208: Numerical Computation for Engineers**
- นักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ที่ลงทะเบียนเรียนในกระบวนวิชา **261405: Advanced Computer Engineering Technology**
- นักศึกษาภาควิชาวิศวกรรมระบบสารสนเทศและเครือข่าย (หลักสูตรนานาชาติ) คณะวิศวกรรมศาสตร์ ที่ลงทะเบียนเรียนในกระบวนวิชา **269400: Advanced Information Systems and Network Technology**

ในการทดสอบระบบกับกลุ่มนักศึกษา ผู้เข้าร่วมได้ทำการใช้งาน ScoreOBE+ ผ่านอุปกรณ์สมาร์ทโฟน คอมพิวเตอร์และแล็ปท็อป โดยมีการประเมินประสิทธิภาพและความสอดคล้องในการเข้าถึงฟังก์ชันหลักของระบบ ซึ่งรวมถึง

- การดูคุณภาพของตนเองและการวิเคราะห์ค่าทางสถิติ
- การตรวจสอบการกระจายคะแนนและแสดงผลในรูปแบบอิสโทแกรม
- การดูผลลัพธ์การเรียนรู้ **Course Learning Outcomes (CLO)**

หลังจากการทดลองใช้งาน อาจารย์ได้ให้ข้อเสนอแนะผ่านผู้พัฒนาโดยตรง เพื่อปรับปรุงและเพิ่มประสิทธิภาพของระบบให้ดียิ่งขึ้น ทั้งในด้านประสบการณ์การใช้งานและการตอบสนองต่อความต้องการที่แท้จริงของผู้สอน

#### 4.4.3 ผลการทดสอบและการวิเคราะห์ข้อมูล

จากการทดสอบการใช้งาน ScoreOBE+ กับกลุ่มตัวอย่างทั้งนักศึกษาและอาจารย์ พบว่าระบบสามารถตอบสนองความต้องการของผู้ใช้ได้เป็นอย่างดี โดยมีการประเมินผลในหลายแบบ รวมถึงข้อเสนอแนะที่เป็นประโยชน์ต่อการพัฒนาในอนาคต

## สิ่งที่ผู้ใช้ชอบ

ผู้ใช้ส่วนใหญ่ให้ความเห็นในเชิงบวกเกี่ยวกับ ScoreOBE+ โดยเน้นไปที่จุดเด่นของระบบ ดังต่อไปนี้

- ความสะดวกในการเข้าถึงคะแนนและสถิติ – นักศึกษาสามารถตรวจสอบคะแนนและวิเคราะห์ผลการเรียนของตนเองได้อย่างง่ายดายผ่านอินเทอร์เฟซที่ใช้งานง่าย
- ระบบแสดงผลข้อมูลที่ชัดเจน – พังก์ชันการแสดงผลค่าทางสถิติ เช่น ค่าเฉลี่ย ค่ามัธยฐาน และค่าความเบี่ยงเบนมาตรฐาน ทำให้ผู้ใช้สามารถเข้าใจการกระจายของคะแนนในระบบวิชาได้ดีขึ้น
- การแสดงผลอิเล็กทรอนิกส์และกราฟการแจกแจงคะแนน – กราฟช่วยให้ผู้ใช้สามารถมองเห็นแนวโน้มของคะแนนได้ชัดเจนและเข้าใจสถานะของตนเองเมื่อเทียบกับกลุ่มเรียนเดียวกัน
- การเข้มข้นของคะแนนกับ CLO และ PLO – ระบบสามารถช่วยให้อาจารย์เข้าใจความสัมพันธ์ระหว่างการประเมินผลและผลการเรียนรู้ของระบบวิชาได้ดีขึ้น
- การนำออกข้อมูลเพื่อส่งให้กับมหาวิทยาลัย – อาจารย์สามารถดาวน์โหลดข้อมูลในรูปแบบที่สะดวกสำหรับการนำไปใช้งานต่อ เช่น การสร้างเอกสาร TQF3, TQF5 และ TQF7
- อินเทอร์เฟซรองรับกับทุกขนาดหน้าจอ – ระบบสามารถใช้งานได้ทั้งบนเดสก์ท็อปและอุปกรณ์พกพาโดยไม่มีปัญหาสำคัญ

## สิ่งที่ผู้ใช้อยากให้ปรับปรุง

แม้ว่าผู้ใช้ส่วนใหญ่จะพึงพอใจกับ ScoreOBE+ แต่ยังมีบางประเด็นที่ต้องได้รับการปรับปรุงประสบการณ์การใช้งาน

- การแสดงคะแนนรวมของระบบวิชา – นักศึกษาต้องการให้ระบบสามารถคำนวณและแสดงคะแนนรวมของตนเองในระบบวิชา เพื่อให้สามารถประเมินผลลัพธ์การเรียนและดูคะแนนรวมได้ง่ายขึ้น
- โหมดกลางคืน (Dark Mode) – มีข้อเสนอให้เพิ่ม Dark Mode เพื่อลดอาการล้าสายตาและ改善ตารางเวลาการใช้งาน
- การออกแบบ UI ให้รองรับการใช้งานบนมือถือดียิ่งขึ้น – นักศึกษาบางส่วนพบว่าการใช้งานผ่านอุปกรณ์พกพาอย่างไม่สะดวกเท่าที่ควร โดยเฉพาะการดูกราฟเนื่องจากต้องหมุนスマาร์ทโฟนให้เป็นแนวนอน (Landscape) และการเปลี่ยนระหว่างหน้าต่างต่าง ๆ ยังต้องอาศัยความเคยชินในการกด Breadcrumb
- การปรับปรุงการแจ้งเตือน – ผู้ใช้ต้องการให้ระบบแจ้งเตือนเมื่อมีการอัปเดตคะแนน เพื่อให้นักศึกษาสามารถติดตามผลการเรียนของตนเองได้แบบเรียลไทม์
- ความชัดเจนของข้อมูล CLO และการเข้มข้นกับการประเมิน – นักศึกษาบางส่วนยังมีข้อสงสัยเกี่ยวกับวิธีการเข้มข้นของ CLO และคะแนนของตนเองว่าคืออะไรให้ธิบายเพิ่มเติม

## สิ่งที่ผู้ใช้ตั้งคำถาม

ในระหว่างการทดสอบ มีข้อสงสัยจากผู้ใช้ที่ควรนำมาพิจารณาเพื่อตอบโจทย์การใช้งานให้ดียิ่งขึ้น

- **เกณฑ์การให้คะแนนของ CLO** – นักศึกษาบางส่วนต้องการทราบว่าแต่ละ CLO มีผลต่อคะแนนรวมของกระบวนการวิชาอย่างไร
- **คำอธิบายของค่าทางสถิติ** – นักศึกษาตั้งข้อสังเกตเกี่ยวกับค่าทางสถิติ เช่น ค่าเฉลี่ยและค่าความเบี่ยงเบนมาตรฐาน หรือค่าควอไทล์ 3 และ 1 ต้องการให้มีคำอธิบายเพิ่มเติม
- **วิธีการเปลี่ยนช่วงคะแนนของ Assessment Tool** – อาจารย์บางท่านมีข้อสงสัยเกี่ยวกับการกำหนดช่วงคะแนนของเครื่องมือประเมิน และต้องการให้มีคำอธิบายเพิ่มเติม หรือคุณมีการใช้งานในการกำหนดช่วงคะแนนว่าเป็นช่วงเปิดหรือปิดตรงจุดคะแนนใดบ้าง (Closed & Opened Interval)
- **การรองรับกระบวนการวิชาที่มีรูปแบบการให้คะแนนที่แตกต่างกัน** – อาจารย์บางท่านสอบถามว่า ระบบสามารถรองรับวิชาที่ใช้การประเมินแบบ S/U หรือวิชาที่ใช้เกณฑ์คะแนนเฉพาะได้หรือไม่

## สิ่งที่ผู้ใช้ให้เป็นไอเดีย

นอกจากข้อเสนอแนะเกี่ยวกับการปรับปรุงระบบ ผู้ใช้งังให้แนวคิดที่เป็นประโยชน์ต่อการพัฒนา Score-OBE+ ในอนาคต

- **การเพิ่มระบบแจ้งเตือน** – นักศึกษาบางส่วนเสนอให้ระบบสามารถแจ้งเตือนเมื่อคะแนนของพากเข้าถูกประ公示 หรือคะแนนของพากเขามีแนวโน้มต่ำกว่าค่าเฉลี่ยของชั้นเรียนเพื่อให้สามารถวางแผนปรับปรุงผลการเรียนได้
- **การพัฒนา Dashboard สรุปผลการเรียน** – ผู้ใช้เสนอให้มีหน้า Dashboard ที่สรุปคะแนนของแต่ละกระบวนการวิชา พร้อมแสดงกราฟเปรียบเทียบคะแนนของผู้ใช้กับค่าเฉลี่ยของทั้งกลุ่มเรียน หรือเทียบกับรุ่นพี่ปีก่อน ๆ
- **การบูรณาการกับแอป CMU Mobile** – นักศึกษาหลายคนแนะนำให้ ScoreOBE+ สามารถเชื่อมโยงกับระบบของมหาวิทยาลัย เพื่อให้สามารถเข้าถึงคะแนนจากแอปเดียว
- **การปรับแต่งอินเทอร์เฟซตามผู้ใช้** – มีข้อเสนอให้เพิ่มฟังก์ชันให้ผู้ใช้สามารถปรับแต่งการแสดงผลของระบบได้ เช่น การเลือกสีของ UI หรือการกำหนดธีมแบบกราฟที่ต้องการ

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

## 5.1 ข้อจำกัดของระบบ

จากการสำรวจความคิดเห็นของนักศึกษาจำนวน 803 คน ตามมหาวิทยาลัยเชียงใหม่ พบร่วมระบบ Score-OBE+ มีบทบาทสำคัญในการช่วยให้นักศึกษาเข้าใจแนวโน้มของคะแนนและผลลัพธ์การเรียนรู้ของตนเองได้ดีขึ้น ซึ่งเป็นประโยชน์อย่างยิ่งในการวางแผนการเรียนและการพัฒนาตนเอง ScoreOBE+ เป็นระบบที่ช่วยบริหารจัดการข้อมูลผลการเรียนรู้ตามกรอบ OBE (Outcome-Based Education) โดยเน้นการเก็บข้อมูลคะแนนและการวิเคราะห์ผลลัพธ์การเรียนรู้ของนักศึกษา ระบบสามารถอ่านวิเคราะห์ผลให้กับผู้ใช้กลุ่มต่างๆ ได้แก่ ผู้ดูแลระบบ ผู้ดูแลหลักสูตร อาจารย์ และนักศึกษา อย่างไรก็ตาม ในการใช้งานระบบยังมีข้อจำกัดบางประการ ดังนี้

- ข้อจำกัดด้านภาษาในระบบ **ScoreOBE+** – ข้อจำกัดเกี่ยวกับการรองรับภาษาอังกฤษเพียงอย่างเดียว ผู้ใช้งานต้องการเลือกภาษาที่ใช้แสดงผลบนระบบ
  - ข้อจำกัดในการประกาศคะแนน – หลังจากที่อาจารย์ได้ประกาศคะแนน ระบบไม่มีการแจ้งเตือนไปยังนักศึกษา ซึ่งข้อจำกัดนี้อาจทำให้นักศึกษาได้รับข้อมูลคะแนนของตนเองที่ล่าช้า
  - ข้อจำกัดในการปรับแต่ง **UI** – การปรับแต่งหน้า Dashboard ผู้ใช้งานต้องการเลือกคูณค่ากำหนดค่าเองในส่วนของแสดงว่าผู้ใช้ต้องการเห็นอะไรแสดงบนหน้าจอหลักบ้าง ซึ่งข้อจำกัดนี้อาจทำให้นักศึกษาได้รับประสบการณ์การใช้งานที่ไม่ดี

ทั้งนี้ จากการทดสอบการใช้งานระบบ และการประเมินการใช้งานจากผู้ใช้งานจริง พบว่าอาจารย์และนักศึกษาสามารถใช้งาน ScoreOBE+ ได้ตรงกับวัตถุประสงค์ที่พัฒนาได้ดีทั้งไว้ทุกข้อ

## 5.2 สรุปผล

นักศึกษาส่วนใหญ่เห็นพ้องกันว่าการนำเสนอข้อมูลแบบกระจายความแนนและ CLO (Course Learning Outcomes) ช่วยให้พวากเข้ามารองเทื่นภาพรวมของการเรียนรู้ของตนเองได้อย่างชัดเจน คะแนนเฉลี่ยที่ได้รับจากหัวข้อประเมินแต่ละด้านมีค่าอยู่ในชั้งสูง สะท้อนถึงประโยชน์และประสิทธิภาพของระบบ ScoreOBE+ โดยสามารถสรุปผลการประเมินได้ดังนี้:

- การเข้าใจภาพรวมของการดูคุณภาพและค่าทางสถิติ: ค่าเฉลี่ย **4.48**  
ระบบช่วยให้นักศึกษาเห็นแนวโน้มของผลการเรียนในระดับมหาวิทยาลัยได้ชัดเจน
  - การกระจายคะแนนช่วยให้เข้าใจความสามารถของตนเอง: ค่าเฉลี่ย **4.49**  
นักศึกษาสามารถประเมินศักยภาพของตนเองได้จากการกระจายคะแนน
  - CLO ช่วยวางแผนการเรียนและเตรียมตัวสำหรับบทเรียน: ค่าเฉลี่ย **4.50**  
CLO เป็นเครื่องมือที่มีประสิทธิภาพในการจัดการแผนการเรียนของนักศึกษา

- การเห็นภาพรวมการกระจายคะแนนของรายวิชาได้ชัดเจน: ค่าเฉลี่ย 4.54

การแสดงผลคะแนนช่วยให้นักศึกษาสามารถประเมินและปรับปรุงแนวทางการเรียนของตนเองได้แม่นยำ

- การกระจายคะแนนช่วยให้ปรับปรุงการเรียนรู้ของตนเอง: ค่าเฉลี่ย 4.41

การใช้ข้อมูลจากการกระจายคะแนนสามารถนำไปสู่การพัฒนาตนเองได้จริง

- CLO** ช่วยให้เข้าใจจุดแข็งและจุดอ่อนของตนเอง: ค่าเฉลี่ย 4.44

นักศึกษาสามารถระบุข้อดีและข้อเสียของตนเองได้ชัดเจน และใช้ข้อมูลนี้เพื่อพัฒนาแผนการเรียนรู้ที่เหมาะสมกับตนเอง

### 5.3 ปัญหาที่พบและแนวทางการแก้ไข

จากการพัฒนาระบบ ScoreOBE+ พับปัญหาสำคัญที่ส่งผลต่อประสิทธิภาพและความคล่องตัวของโครงการดังนี้:

- ความไม่แน่นอนของข้อกำหนด (Requirement Instability)** – ข้อกำหนดของระบบมีการเปลี่ยนแปลงบ่อยครั้ง ส่งผลให้การออกแบบระบบต้องมีการปรับเปลี่ยนตาม ทำให้การพัฒนาล่าช้า
- ขอบเขตของโครงการที่ขยายตัว (Expanding Scope)** – ในระหว่างการพัฒนา พบว่าขอบเขตของโครงการขยายกว้างขึ้นจากการรับรวมข้อกำหนดเพิ่มเติมอย่างต่อเนื่อง ทำให้ต้องมีการปรับเปลี่ยนแผนการทำงานตลอดเวลา
- ข้อจำกัดด้านการออกแบบ UI (UI Feasibility)** – การออกแบบอินเทอร์เฟซบางส่วนไม่สามารถนำไปใช้จริงได้ ทำให้ต้องมีการปรับเปลี่ยนอยู่เสมอเพื่อให้สามารถใช้งานได้จริงและตอบโจทย์ผู้ใช้
- ขาดความเข้าใจเชิงลึกเกี่ยวกับระบบ (Limited Knowledge)** – ทีมพัฒนานี้มีข้อมูลเกี่ยวกับระบบไม่มากพอ ส่งผลให้การออกแบบและพัฒนาต้องมีการเปลี่ยนแปลงบ่อยครั้ง ซึ่งกระทบต่อประสิทธิภาพของกระบวนการพัฒนา

เพื่อแก้ไขปัญหาเหล่านี้ ควรมีการกำหนดข้อกำหนดให้แน่นอนตั้งแต่เริ่มต้น ใช้แนวทาง Agile ในการพัฒนา เพื่อลดผลกระทบจากการเปลี่ยนแปลง

### 5.4 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

เพื่อให้ระบบ ScoreOBE+ มีประสิทธิภาพมากขึ้นในอนาคต ควรดำเนินการตามแนวทางดังต่อไปนี้:

- ปรับปรุงความเสถียรของข้อกำหนดโดยการวางแผนและกำหนดขอบเขตโครงการให้ชัดเจนตั้งแต่เริ่มต้น
- ใช้แนวทาง Agile ในการพัฒนาเพื่อลดผลกระทบจากการเปลี่ยนแปลงข้อกำหนด
- พัฒนา UI ให้สามารถปรับแต่งได้มากขึ้นตามความต้องการของผู้ใช้งาน
- เสริมฟังก์ชันการแจ้งเตือนและการวิเคราะห์ข้อมูลให้ครอบคลุมมากขึ้น เพื่อช่วยให้นักศึกษาได้รับข้อมูลที่เป็นประโยชน์ต่อการเรียน
- พัฒนาฟังก์ชัน Skill เชื่อมโยงกับ CLO ให้ครอบคลุม OBE Framework มากขึ้น

## 5.5 สรุปผล

จากการสำรวจนี้สามารถสรุปได้ว่า **ScoreOBE+** เป็นเครื่องมือที่มีประสิทธิภาพในการช่วยให้นักศึกษาเข้าใจผลการเรียนรู้ของตนเองได้ดีขึ้น และสามารถนำข้อมูลที่ได้รับไปใช้ในการปรับปรุงและพัฒนาการเรียนรู้ได้อย่างเป็นระบบ ซึ่งจะส่งผลต่อการพัฒนาผลสัมฤทธิ์ทางการเรียนในระยะยาว

## บรรณานุกรม

- [1] Api คืออะไร? ทำไมถึงมีทุกวัน? อธิบายแบบเข้าใจง่าย, 2023. <https://www.dmit.co.th/th/zendesk-updates-th/what-is-api/>. Accessed: 2025-03-17.
- [2] Chart.js. <https://www.chartjs.org/docs/latest/>. Accessed: 2025-03-20.
- [3] Docker คืออะไร: คุณมีเริ่มต้นพร้อมรายละเอียดทางเทคนิค. <https://www.askme.co.th/article/what-is-docker/>. Accessed: 2025-03-17.
- [4] Win Eiwwongcharoen. ลองเล่น framer motion (ตอนที่ 1): เริ่มต้นทำ animation ง่ายๆ บน react ด้วย framer motion, 2020. <https://medium.com/@rennerwin/ลอง-เล่น-framer-motion-ตอน-ที่-1-เริ่ม-ต้นทำ-animation-ง่ายๆ-บน-react-ด้วย-framer-motion-bb0f68fc9d40>. Accessed: 2025-03-20.
- [5] รู้จัก 'figma' tool ออกแบบ web/app ที่ดีไซน์เนอร์ทั่วโลกเลือกใช้, 2023. <https://www.truedigitalacademy.com/blog/get-to-know-figma>. Accessed: 2025-03-17.
- [6] Gestalt principle in ui design ทฤษฎีทางจิตวิทยาที่อาจทำให้การมองสิ่งต่างๆของคุณเปลี่ยนไป. <https://medium.com/the-existing/gestalt-principle-in-uidesign-33ee34ae8f21>. Accessed: 2024-02-29.
- [7] Exploring the gestalt principles of design. <https://www.toptal.com/designers/ui/gestalt-principles-of-design>. Accessed: 2024-02-29.
- [8] Interaction design principle feedback (หลักการออกแบบการโต้ตอบ ข้อเสนอแนะ) คืออะไร? <https://medium.com/upskill-ux/interaction-design-principle-feedback-หลัก-การ-ออกแบบ-การ-โต้ตอบ-ข้อ-เสนอ-แนะ-คือ-อะไร-11b3e6598e44>. Accessed: 2024-02-29.
- [9] Boost your ux with these successful interaction design principles. <https://www.toptal.com/designers/interactive/interaction-design-principles>. Accessed: 2024-02-29.
- [10] Supreecha Jaijumpa. ไอเดียใช้ framer motion เพิ่มลูกเล่น animation ให้เว็บไซต์, 2023. <https://blog.pdkm.tech/ไอเดียใช้-framer-motion-เพิ่ม-ลูกเล่น-animation-ให้-เว็บไซต์-9a195fb34b51>. Accessed: 2025-03-20.
- [11] Jakob law (เจ้า定律) คืออะไร? <https://medium.com/upskill-ux/jakob-law-เจ้า定律-53af1ef53052>. Accessed: 2024-02-29.
- [12] Ux laws: Jakob's law explained in simple terms. <https://bootcamp.uxdesign.cc/ux-laws-jakobs-law-explained-in-simple-terms-fbeb0d794318>. Accessed:

2024-02-29.

- [13] Mantine. <https://mantine.dev/>. Accessed: 2025-03-20.
- [14] Mongodb คืออะไร. <https://appmaster.io/th/blog/mongodb-khuue-aair>. Accessed: 2024-02-29.
- [15] Mongoose คือ อะไร ดี อย่างไร ใช้งาน อะไร ได้ บ้าง ข้อ เสีย มี อะไร บ้าง. [https://expert-programming-tutor.com/tutorial/article/KE000012\\_What\\_is\\_Mongoose\\_What\\_can\\_be\\_used\\_What\\_are\\_the\\_disadvantages.php](https://expert-programming-tutor.com/tutorial/article/KE000012_What_is_Mongoose_What_can_be_used_What_are_the_disadvantages.php). Accessed: 2025-03-20.
- [16] Kanthima Muangsen. ทำไมถึงต้องใช้ typescript ทั้งที่มี javascript อยู่แล้ว ?, 2021. <https://www.borntodev.com/2021/05/13/ทำ-ไม่ถึง-ต้อง-ใช้-typescript/>. Accessed: 2025-03-20.
- [17] แนวคิดของ mvc design pattern. <https://devhub.in.th/blog/mvc-design-pattern>. Accessed: 2024-02-29.
- [18] Andrew Myers. Some useful software. <https://www.cs.cornell.edu/andru/useful.html>. Accessed: 2024-11-21.
- [19] ทำความรู้จัก nest js และวิธีการใช้!, 2020. <https://kritsadapk.medium.com/ทำความรู้จัก-nest-js-และวิธีการใช้-6a498bef8b01>. Accessed: 2025-03-17.
- [20] Papimpat Nimprasert. ทำความรู้จักกับ react และการใช้งานเบื้องต้น, 2024. <https://www.borntodev.com/2024/05/13/ทำความรู้จักกับ-react/>. Accessed: 2025-03-17.
- [21] ทำไมต้อง obe? <https://www.kmutt.ac.th/education/>. Accessed: 2024-02-29.
- [22] เรียนรู้การใช้ jwt กับ passport.js authentication, 2019. <https://siriphonnot.medium.com/เรียนรู้การใช้-jwt-กับ-passport-js-authentication-7d5bde58bd22>. Accessed: 2025-03-20.
- [23] ลอง gen pdf กันจาก pdfkit ใน nodejs, 2018. <https://wisdomgoody.medium.com/ลอง-gen-pdf-กัน-จาก-pdfkit-ใน-nodejs-e11d71dc2521>. Accessed: 2025-03-20.
- [24] Chai Phonbopit. Redux คืออะไร? + เริ่มต้นเรียนรู้ redux ร่วมกับ react กันดีกว่า, 2018. <https://www.devahoy.com/blog/2018/07/introduction-to-redux/>. Accessed: 2025-03-20.

- [25] Postman : มาเร็วจักกับ postman กันเถอะ. <https://medium.com/lotuss-it/postman-มา-เร็วจักกับ-postman-กัน-แล้ว-157193f5a215>. Accessed: 2025-03-17.
- [26] มาทำความรู้จัก procreate และวิเคราะห์ใน ipad สำหรับมือใหม่, 2024. <https://www.trueplookpanya.com/knowledge/content/94253-itapp-it->. Accessed: 2025-03-17.
- [27] [beginner] มาทำความรู้จักกับ rest api กันดีกว่า (มีตัวอย่าง). <https://medium.com/@settawatjanpuk/beginner-มา-ทำความ-รู้จัก-กับ-rest-api-กัน-ดี-กว่า-มี-ตัว-อย่าง-c855d93beaf3>. Accessed: 2024-02-29.
- [28] Swagger คืออะไรและใช้อย่างไร. <https://appmaster.io/university/th/tutorials/endpoints/swagger-khuue-aairaelaaich-yaangair>. Accessed: 2025-03-20.
- [29] User-centered design. [https://en.wikipedia.org/wiki/User-centered\\_design](https://en.wikipedia.org/wiki/User-centered_design). Accessed: 2024-02-29.
- [30] แนะนำ user-centered design canvas เครื่องมือ ux ที่ทำให้เข้าใจผู้ใช้มากขึ้น. <https://medium.com/the-existing/แนะนำ-user-centered-design-canvas-เครื่องมือ-ux-ที่ทำให้เข้าใจผู้ใช้มากขึ้น-bd9958a4b998>. Accessed: 2024-02-29.

ภาคผนวก

## ภาคผนวก ก คู่มือการใช้งานระบบ

การใช้งานระบบและฟังก์ชันการทำงานต่าง ๆ จะแบ่งตามบทบาทของผู้ใช้งาน

### ก.1 ผู้สอน (Instructor)

ผู้สอนมีสิทธิ์และความสามารถในการเข้าถึงและจัดการเนื้อหาการเรียนการสอนภายในระบบ Score-OBE+ โดยบทบาทของผู้สอนในระบบจะช่วยให้สามารถจัดการเอกสาร TQF3 TQF5 และผลการเรียนของนักศึกษา สิทธิ์ของผู้สอนประกอบด้วยฟังก์ชันหลักดังต่อไปนี้:

#### ก.1.1 การจัดทำเอกสารประกอบการเรียนการสอน

ผู้สอนสามารถสร้างและแก้ไขเอกสารประกอบการเรียนการสอนที่เกี่ยวข้องกับการประเมินผลการเรียนรู้ เช่น มคอ.3 (TQF3) และ มคอ.5 (TQF5) ซึ่งเป็นเอกสารสำคัญที่ใช้ในระบบการศึกษาตามกรอบมาตรฐาน คุณวุฒิระดับอุดมศึกษาแห่งชาติ โดย:

- มคอ.3 (TQF3) เป็นเอกสารที่ใช้สำหรับวางแผนการเรียนการสอนและการประเมินผลของกระบวนวิชา
- มคอ.5 (TQF5) เป็นเอกสารที่ใช้สำหรับรายงานผลการเรียนรู้ของนักศึกษาหลังจากสิ้นสุดการสอน

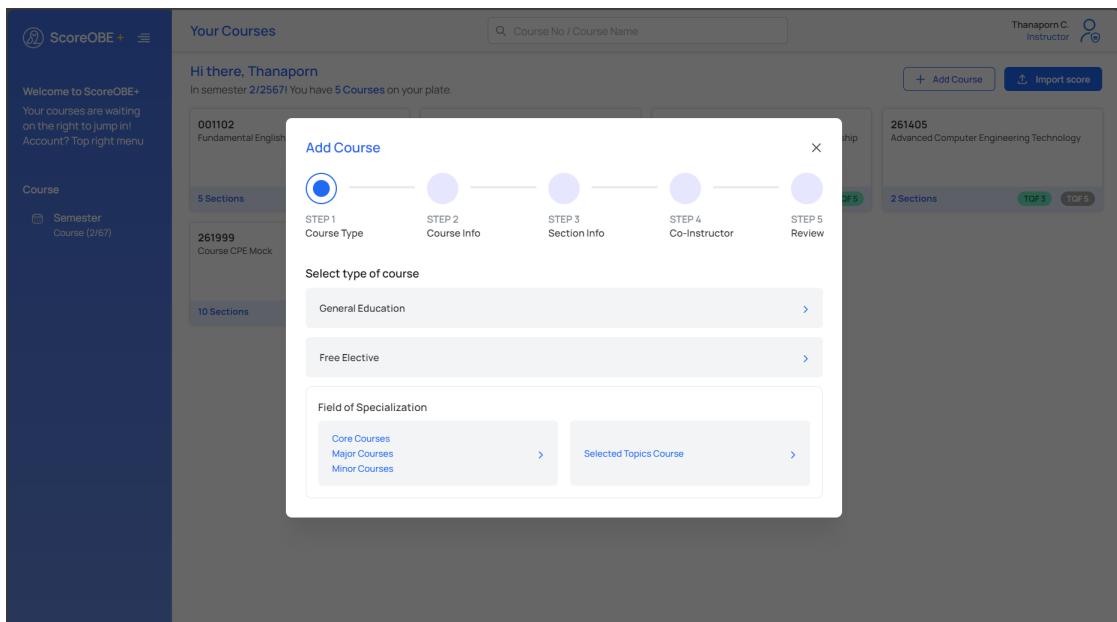
ทั้งนี้ ผู้สอนต้องดำเนินการกรอกข้อมูลใน มคอ.3 (TQF3) ให้เสร็จสิ้นก่อน จึงจะสามารถแก้ไขและจัดทำเอกสาร มคอ.5 (TQF5) ได้ นอกจากนี้ ระบบยังเปิดโอกาสให้ผู้สอนนำเข้าข้อมูลค่าคะแนนของนักศึกษาเพื่อช่วยจัดทำ มคอ.5 (TQF5) ให้มีความละเอียดและสะท้อนผลการเรียนได้อย่างแม่นยำยิ่งขึ้น อย่างไรก็ตาม หากผู้สอนไม่มีค่าคะแนนของนักศึกษาอยู่ในระบบ ScoreOBE+ ก็สามารถเลือกดำเนินการจัดทำ มคอ.5 (TQF5) ได้ด้วยตนเอง (manual) โดยป้อนข้อมูลทั้งหมดเอง เช่นเดียวกับระบบ CPE QA

#### ก.1.2 การจัดการกระบวนวิชาและการนำเข้าค่าคะแนน

ผู้สอนสามารถจัดการกระบวนวิชาและนำเข้าค่าคะแนนของนักศึกษาได้ผ่านระบบ ScoreOBE+ โดยสามารถดำเนินการดังต่อไปนี้:

- สร้างกระบวนวิชาใหม่ ผู้สอนสามารถสร้างกระบวนวิชาใหม่และกำหนดรายละเอียดต่าง ๆ เช่น รหัสวิชา ชื่อกระบวนวิชา เลขกลุ่มเรียน หลักสูตรที่เชื่อมโยง และผู้สอนร่วมในแต่ละกลุ่มเรียน เมื่อผู้สอนสร้างกระบวนวิชา ระบบจะแสดง Modal ให้กรอกข้อมูลใน 5 ขั้นตอน:

## ขั้นตอนที่ 1: เลือกประเภทกระบวนการวิชา

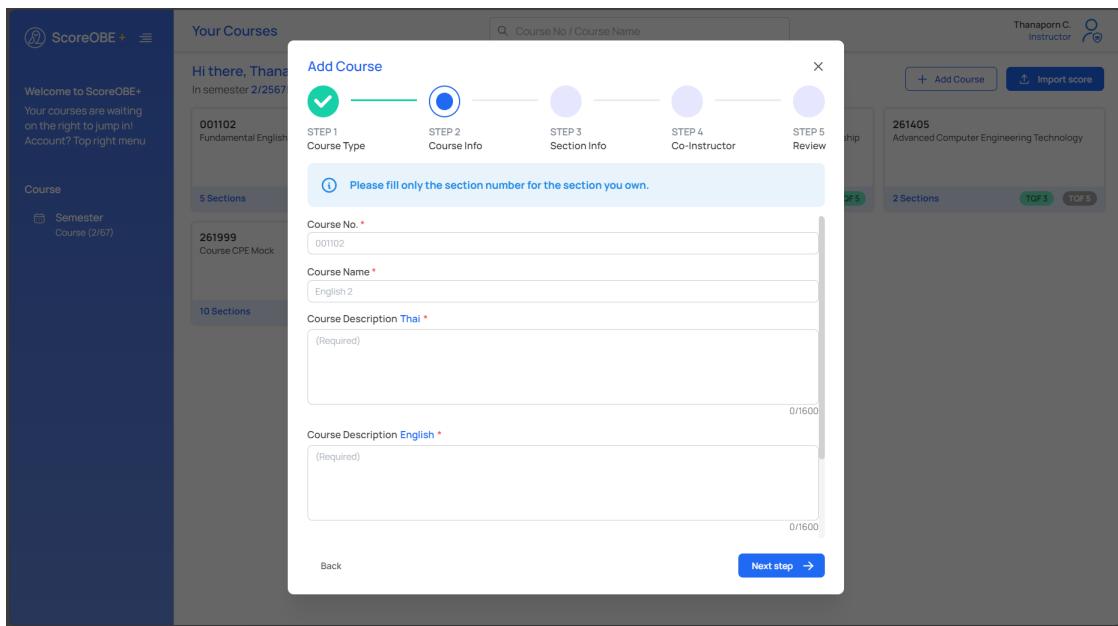


รูปที่ ก.1: ขั้นตอนที่ 1 เลือกประเภทกระบวนการวิชา

ในขั้นตอนแรกของการเพิ่มกระบวนการวิชา ผู้ใช้ต้องเลือกประเภทของกระบวนการวิชาที่ต้องการเปิดสอน โดยประเภทของกระบวนการวิชา มีทั้งหมด 4 ประเภท ได้แก่:

- **General Education** – วิชาศึกษาทั่วไปที่เป็นพื้นฐานสำหรับทุกสาขาวิชา
- **Free Elective** – วิชาเลือกเสรีที่สามารถเลือกลงทะเบียนได้โดยไม่มีข้อจำกัดด้านสาขาวิชา
- **Field of Specialization** – วิชาเฉพาะทางที่เป็นส่วนหนึ่งของหลักสูตรเฉพาะ
- **Selected Topics** – วิชาที่มีเนื้อหาพิเศษหรือหัวข้อเฉพาะที่อาจมีการเปลี่ยนแปลงตามภาคการศึกษา

## ขั้นตอนที่ 2: กรอกรายละเอียดของกระบวนวิชา



The screenshot shows the 'Add Course' interface on the ScoreOBE+ platform. The process is divided into five steps: STEP 1 Course Type, STEP 2 Course Info, STEP 3 Section Info, STEP 4 Co-Instructor, and STEP 5 Review. The current step is STEP 2 Course Info. A note at the top of the form says: 'Please fill only the section number for the section you own.' The form fields are as follows:

- Course No. \*: 001102
- Course Name \*: English 2
- Course Description Thai \*:  
(Required)
- Course Description English \*:  
(Required)

At the bottom of the form are 'Back' and 'Next step' buttons.

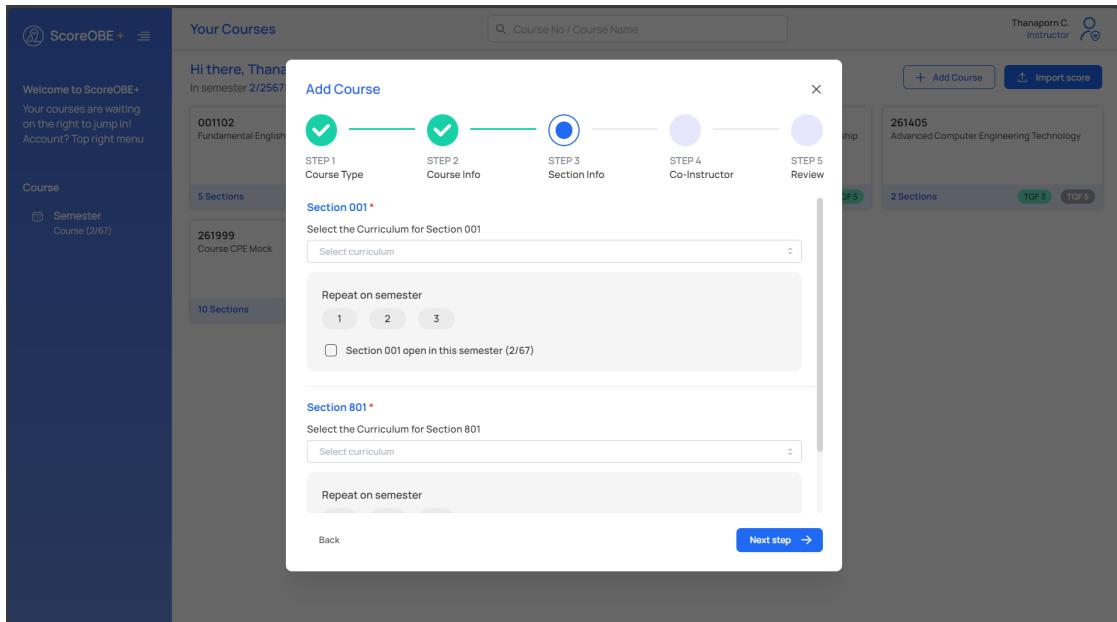
รูปที่ ก.2: ขั้นตอนที่ 2 กรอกรายละเอียดของกระบวนวิชา

หลังจากเลือกประเภทกระบวนวิชาแล้ว ผู้ใช้ต้องกรอกรายละเอียดของกระบวนวิชา ได้แก่:

- รหัสวิชา (Course Code)
- ชื่อวิชา (Course Name)
- คำอธิบายรายวิชา (Course Description)
- เลขกลุ่มเรียน (Section Number)

ข้อมูลเหล่านี้มีความสำคัญต่อการบริหารจัดการหลักสูตรและการลงทะเบียนของนักศึกษา

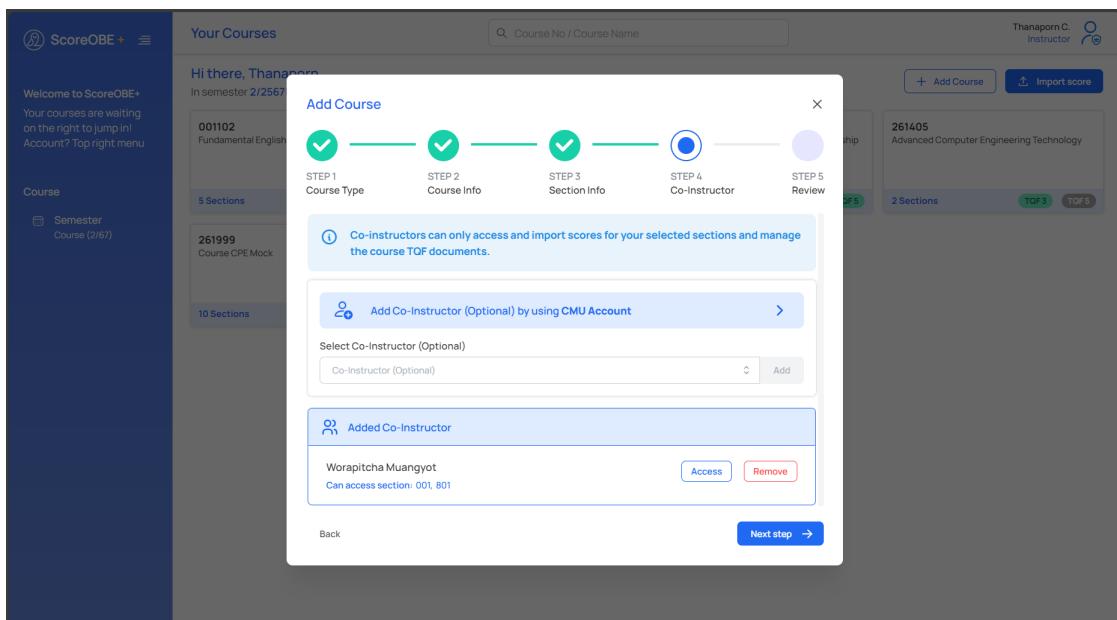
### ขั้นตอนที่ 3: กรอกรายละเอียดของกลุ่มเรียน



รูปที่ ก.3: ขั้นตอนที่ 3 กรอกรายละเอียดของกลุ่มเรียน

ในขั้นตอนนี้ ผู้สอนต้องระบุหลักสูตรที่เกี่ยวข้องกับกลุ่มเรียนที่ได้ระบุในขั้นตอนที่ 2 และกำหนดภาคการศึกษาที่จะเปิดสอน

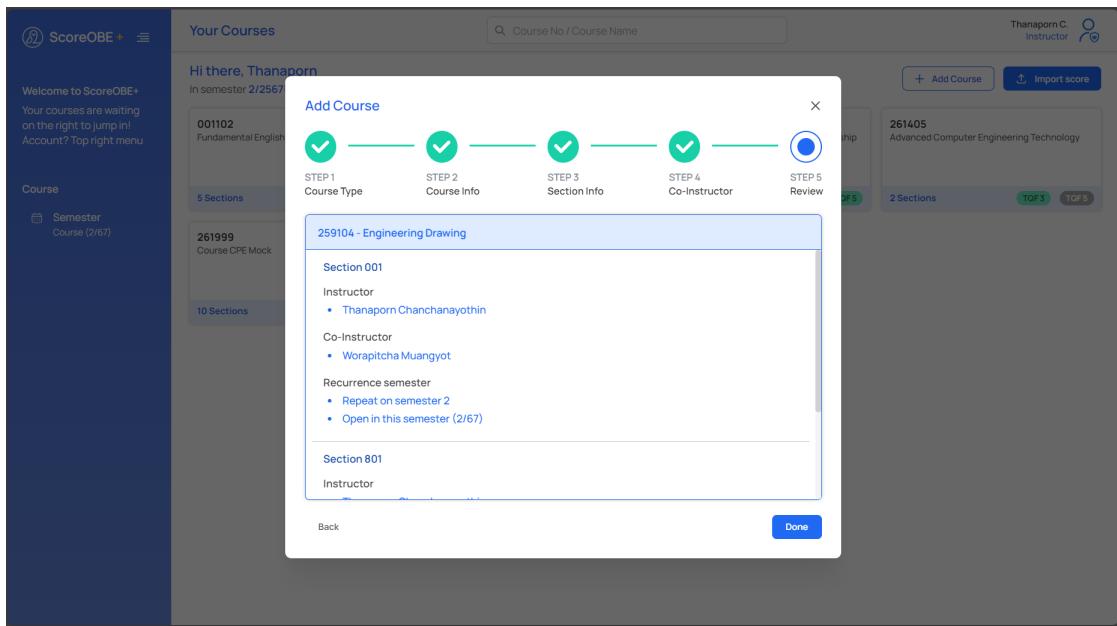
### ขั้นตอนที่ 4: กรอกรายละเอียดของผู้สอนร่วม



รูปที่ ก.4: ขั้นตอนที่ 4 กรอกรายละเอียดของผู้สอนร่วม

หากกระบวนการวิชาไม่ผู้สอนมากกว่าหนึ่งคน ผู้ใช้สามารถเพิ่มผู้สอนร่วมได้โดยเลือกจาก **Dropdown Menu** หรือใช้ **CMU Account** จากนั้นต้องกำหนดเลขกลุ่มเรียนที่ให้สิทธิ์ผู้สอนร่วมในการเข้าถึงคะแนนของนักศึกษา

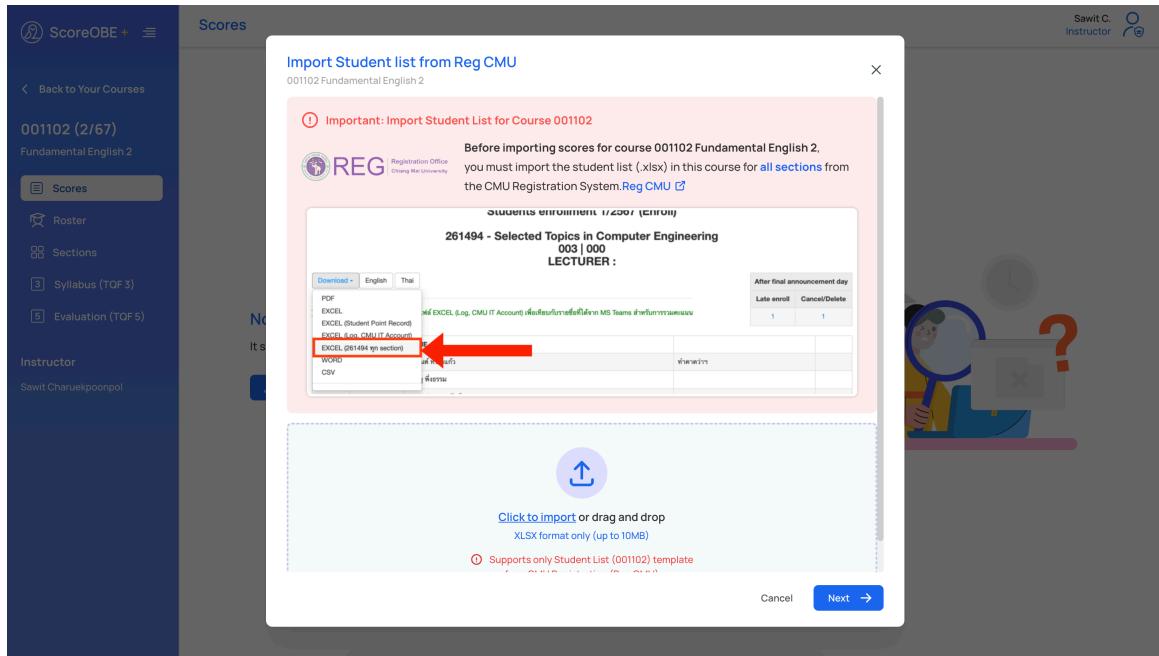
### ขั้นตอนที่ 5: ตรวจสอบความถูกต้องของกระบวนการวิชา



รูปที่ ก.5: ขั้นตอนที่ 5 ตรวจสอบความถูกต้องของกระบวนการวิชา

## 2. นำเข้ารายชื่อนักศึกษา

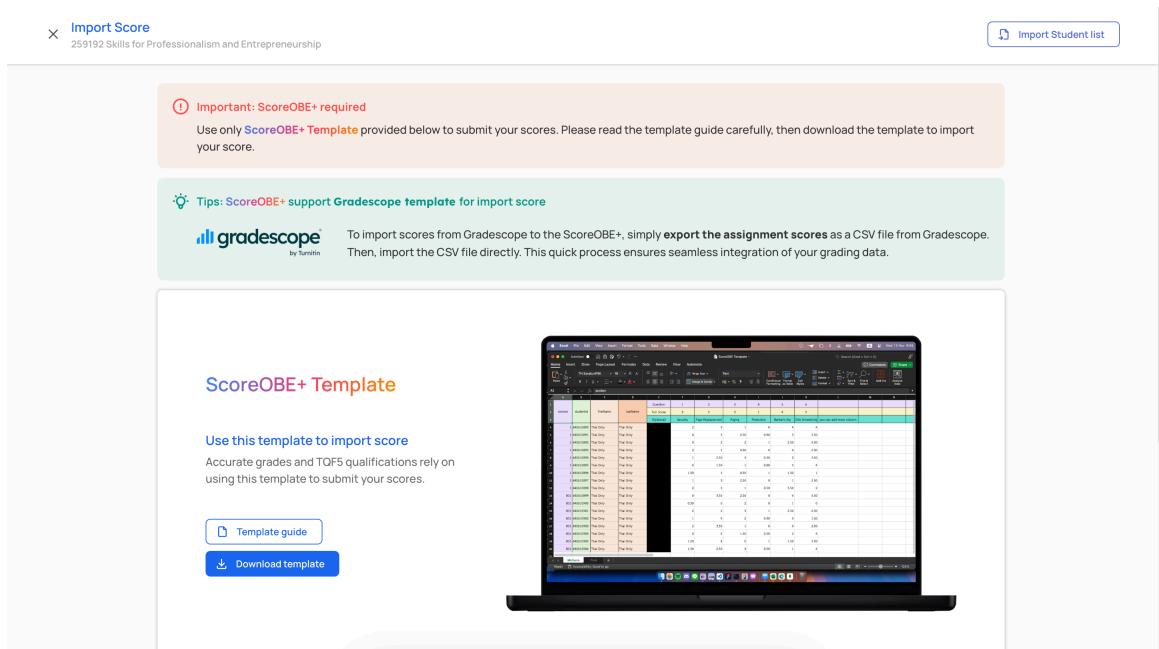
ก่อนนำเข้าคะแนน ระบบจะกำหนดให้ผู้ใช้ต้องนำเข้ารายชื่อนักศึกษาตามแบบเพลตจากสำนักทะเบียนและประมวลผล มหาวิทยาลัยเชียงใหม่เท่านั้น หากเลขกลุ่มเรียนที่สร้างในระบบไม่ตรงกับข้อมูลในแบบเพลต ระบบจะแจ้งเตือนก่อนนำเข้ารายชื่อนักศึกษา และหากพบว่าเลขกลุ่มเรียนในรายชื่อนักศึกษาไม่ตรงกับที่กำหนดไว้ในระบบ รายชื่อนักศึกษาที่ใช้เลขกลุ่มเรียนนั้นจะไม่ถูกนำเข้า



รูปที่ ก.6: Modal นำเข้ารายชื่อนักศึกษา ก่อนนำเข้าคะแนน

### 3. นำเข้าคะแนนนักศึกษา

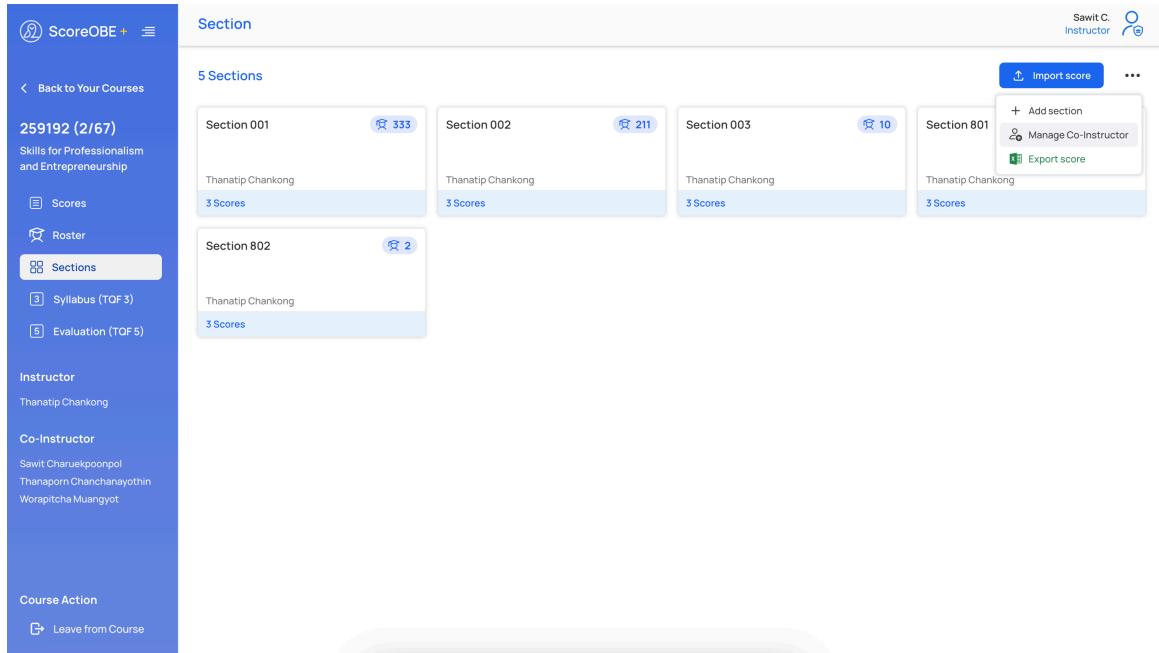
ระบบรองรับการนำเข้าคะแนนในรูปแบบไฟล์สเปรตชีต (Spreadsheet) โดยสามารถใช้ไฟล์เพลตของ ScoreOBE+ และ Gradescope ซึ่งช่วยให้ผู้สอนนำเข้าคะแนนจากแพลตฟอร์มภายนอกได้อย่างสะดวก หากคะแนนนักศึกษามีค่าที่ไม่ถูกต้อง ระบบจะแจ้งเตือนไม่ให้นำเข้าคะแนน และบอกรหัสที่ต้องแก้ไข และหากพบว่าคะแนนนักศึกษารายชื่อตกลงไม่ตรงกับ Course Roster ระบบจะแจ้งเตือนและยืนยันก่อนนำเข้าคะแนน



รูปที่ ก.7: Full Modal นำเข้าคะแนนรายวิชา 259192

#### 4. เพิ่มผู้สอนร่วม (Co-Instructor)

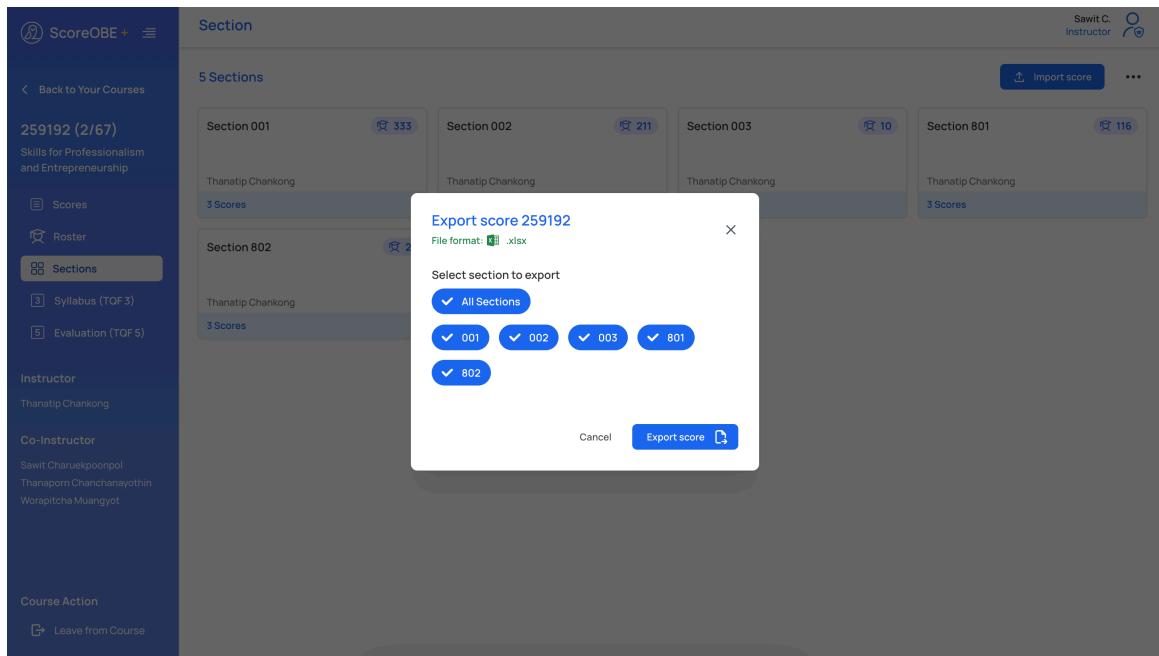
หากกระบวนวิชา มีผู้สอนมากกว่าหนึ่งคน และต้องการเพิ่มผู้สอนร่วมในภายหลัง ผู้สอนสามารถเพิ่ม Co-Instructor พร้อมกำหนดสิทธิ์ให้เข้าถึงข้อมูลคะแนนของบางกลุ่มเรียนตามที่กำหนดได้



รูปที่ ก.8: เมนูรายการเพิ่มผู้สอนร่วม

#### 5. นำออก (Export) คะแนน

ผู้สอนสามารถดาวน์โหลดคะแนนของนักศึกษาในรูปแบบไฟล์ .xlsx เพื่อนำไปวิเคราะห์ข้อมูลเพิ่มเติมหรือใช้จัดทำเอกสารที่เกี่ยวข้อง

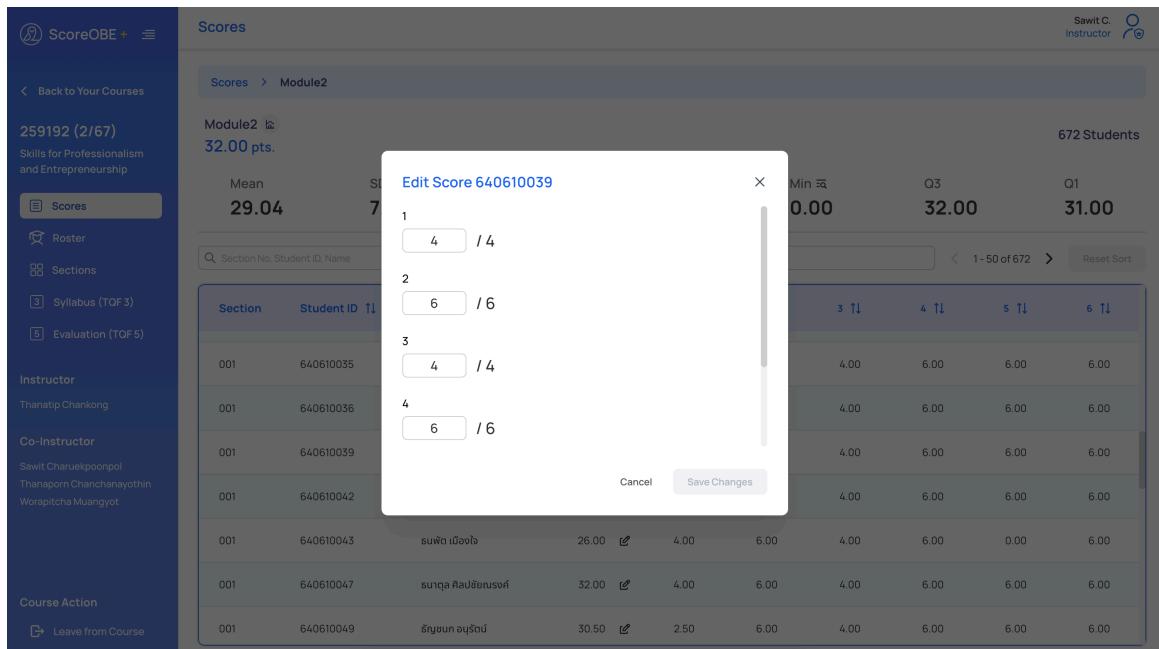


รูปที่ ก.9: Modal เลือก Section ที่ต้องการนำออกคะแนน

### ก.1.3 การจัดการคะแนนและการประมวลผลการเรียน

ภายในระบบ ScoreOBE+ ผู้สอนสามารถจัดการกับคะแนนของนักศึกษาในระบบวิชาที่ตนเองรับผิดชอบได้ โดยสามารถดำเนินการต่าง ๆ ดังต่อไปนี้:

1. **แก้ไขคะแนน** – ผู้สอนสามารถแก้ไขคะแนนของนักศึกษาได้เป็นรายบุคคล หรือแก้ไขคะแนนสำหรับงานมอบหมายแต่ละชิ้นในระบบวิชา



รูปที่ ก.10: Modal แก้ไขคะแนนนักศึกษา

2. ลบคะแนน – ผู้สอนสามารถลบคะแนนที่เคยนำเข้าไปในระบบได้ โดยเลือกที่จะลบคะแนนของงานมอบหมายได้งานมอบหมายหนึ่ง

Score name	Full Scores	Mean	Created	Student(s)	Published
Module2	32	29.04	17 Mar 2025	672	
Module3				672	
Module1				672	

รูปที่ ก.11: Popup ยืนยันการลบคะแนนงานมอบหมาย

3. ประกาศคะแนน – ผู้สอนสามารถประกาศคะแนนของนักศึกษาในระบบได้ สามารถเลือกประกาศคะแนนสำหรับทุกกลุ่มเรียนหรือเฉพาะบางกลุ่มเรียน และเลือกประกาศทุกงานมอบหมายหรือเฉพาะบางงานมอบหมายได้ตามต้องการ โดยหลังจากการประกาศคะแนนนักศึกษาจะเห็นการแสดงผลคะแนนของตนเอง ดังนี้:

- คะแนนพร้อมค่าทางสถิติ เช่น ค่าเฉลี่ย ค่ามัธยฐาน และค่าความเบี่ยงเบนมาตรฐาน เพื่อให้นักศึกษาเข้าใจการกระจายตัวของคะแนนในกระบวนการวิชา
- กราฟการกระจายตัวของคะแนนและฮิสโตแกรม เพื่อให้นักศึกษาปรับเทียบคะแนนของตนเองกับเพื่อนร่วมชั้น

รูปที่ ก.12: Modal เลือกงานมอบหมายที่จะประกาศ

#### ก.1.4 การจัดการเอกสาร TQF3 และ TQF5

ผู้สอนสามารถเข้าถึงและแก้ไขเอกสาร TQF3 (รายละเอียดกระบวนการวิชา) และ TQF5 (รายงานผลการสอน) ของกระบวนการวิชาที่ตนเองสร้าง หรือเป็นผู้สอนร่วม

รูปที่ ก.13: TQF3 Part 1 Course Information

ScoreOBE

Back to Your Courses

261405 (2/67)  
Advanced Computer Engineering Technology

Scores Roster Sections Syllabus (TQF 3) Evaluation (TQF 5)

Instructor  
Dome Potikanond

Co-Instructor  
Kasemait Teeyapan Sawit Charuekiponpol Thanaporn Chanchanayothin Worapitcha Muangyot

Course Action  
Leave from Course

TQF 3

Part 1 Part 2 Part 3 Part 4 Part 5 Part 6 Part 7

Part 2 - คัดลอกข้อมูลของวิชาและแผนการสอน  
Description and Planning

การสอนและประเมินผล \* Evaluation

หลักสูตรของวิชา \* Teaching Method

บรรยาย (Lecture)  
 ปฏิบัติการ (Laboratory)  
 ฝึกปฏิบัติ (Practice)  
 แก้ไขร่วมกัน (Co-operative)

การวัดและประเมินผล \* Evaluation

A-F S/U P

ผลลัพธ์การเรียนรู้ของวิชา (Course Learning Objective: CLO)\*

+ Add CLO

CLO No.	CLO Description	Learning Method	Action
1	สามารถออกแบบซอฟต์แวร์และระบบสารสนเทศที่ตอบสนองความต้องการของผู้ใช้ส่วนได้ส่วนเสียได้อย่างเหมาะสม Able to design a software and information system according to stakeholders requirements.	บรรยาย (Lecture) ปฏิบัติการ (Laboratory)	
2	สามารถพัฒนาและทดสอบการเรียนรู้ของเครื่องห้องสมุดของวิชาได้ Able to develop and to test a machine learning model according to requirements.	บรรยาย (Lecture) ปฏิบัติการ (Laboratory)	
-	สามารถออกแบบและสร้างซอฟต์แวร์และระบบสารสนเทศที่ตอบสนองความต้องการของผู้ใช้ส่วนได้ส่วนเสียได้อย่างเหมาะสม		

Saved: Nov 18, 2024 13:41

รูปที่ ก.14: TQF3 Part 2 Description and Planning

ScoreOBE

Back to Your Courses

261405 (2/67)  
Advanced Computer Engineering Technology

Scores Roster Sections Syllabus (TQF 3) Evaluation (TQF 5)

Instructor  
Dome Potikanond

Co-Instructor  
Kasemait Teeyapan Sawit Charuekiponpol Thanaporn Chanchanayothin Worapitcha Muangyot

Course Action  
Leave from Course

TQF 3

Part 1 Part 2 Part 3 Part 4 Part 5 Part 6 Part 7

Part 3 - ประเมินผลและแผนการสอนวิชา  
Course Evaluation

การกำหนดเกรด (Grading) \*

แบบอ้างอิงคุณภาพ (Norm-Referenced Grading)  
 แบบอ้างเกณฑ์ (Criterion-Referenced Grading)  
 แบบอ้างเกณฑ์และอ้างคุณภาพ (Criterion and Norm-Referenced Grading)

Evaluation Items \*

The total of all methods in the course syllabus must add up to 100%.

+ Add Evaluation Method

No.	Method	Description	Evaluate	Action
1	รายงานฉบับสมบูรณ์ Final report	- เรียนเนย	30%	
2	การเข้าชั้นเรียนและการร่วมมือในการเรียน Class attendance and collaboration	-	10%	
-	การนำเสนอผลงาน	-	20%	

Saved: Nov 18, 2024 14:31

รูปที่ ก.15: TQF3 Part 3 Course Evaluation

CLO Description / Evaluation Item	รายงานฉบับสมบูรณ์ Final report	การเข้าชั้นเรียนและการเข้าห้องเรียน Class attendance	การนำเสนอภาคภาษา Final Presentation	การเรียนรู้ผ่านเครื่องมือออนไลน์ Online self-learning	รายละเอียดโครงการ Progress	Total CLO (%)
<b>CLO-1</b> สามารถออกแบบซอฟต์แวร์และระบบสารสนเทศที่ตอบสนองความต้องการของผู้ใช้งานได้รับเชิงเสียงประทับใจและสนับสนุน Able to design a software and information system according to stakeholders requirements.	7%	0%	4%	0%	7%	18%
<b>CLO-2</b> สามารถพัฒนาแบบจำลองด้านการเรียนรู้ของเครื่องที่ตอบสนองสิ่งแวดล้อมและทดสอบการทำงานได้ Able to develop and to test a machine learning model according to requirements.	4%	0%	0%	0%	4%	8%
<b>CLO-3</b> สามารถออกแบบเครือข่ายคอมพิวเตอร์ขององค์กรที่ตอบสนองความต้องการของผู้ใช้งานได้รับเชิงเสียงประทับใจและสนับสนุน Able to design a campus network according to stakeholders requirements.	8%	0%	4%	0%	8%	20%
<b>Total Assessment (%)</b>	30 / 30%	10 / 10%	20 / 20%	10 / 10%	30 / 30%	<b>100%</b>

รูปที่ ก.16: TQF3 Part 4 Assessment Mapping

ชื่อเอกสาร Main Reference	Description
ใบงานประกอบการเรียนการสอน (slide presentation)	

เอกสารแบบบันทึก Recommended Documents e.g. Lecture notes, E-documents, etc.	Description
เอกสารประกอบการและเอกสารอธิบายที่เกี่ยวข้อง	

รูปที่ ก.17: TQF3 Part 5 Course Materials

Part 6 - กรรมการประเมินรวมวิชาและกระบวนการปรับปรุง  
Course Review and Improvement Processes

ทั้งหมด 7 รายการ

1. Strategies for evaluating course effectiveness by students

- ไม่มี (None)
- แบบประเมินรวมวิชา  
(Course assessment form)
- กรณีกลุ่ม: วิชาชีวะและสุขภาพ  
(Instructor-student group discussion)
- กรณีกลุ่ม: จิตวิทยาและมนุษย์  
(Student behavior reflection)
- ข้อเสนอแนะผ่านบอร์ดอินสแตนซ์ที่อาจารย์ได้อธิบายให้เป็นอีกทางการสื่อสารกับนักศึกษา  
(Suggestions through the instructor's webboard for student communication.)

Saved: Nov 18, 2024 13:50 | Save Part 6

รูปที่ ก.18: TQF3 Part 6 Course Review and Improvement Processes

Part 7 - กรณีของหัวข้อประเมินวัตถุประสงค์การเรียนรู้  
Curriculum Mapping

CPE-2563

CLO Mapping \*

Section 001, 801

Each CLO must be linked to at least one PLO. And if you see \* in a PLO column, at least one of your CLOs must be linked to that required PLO

CLO Description (8 CLOs)	PLO-1	PLO-2	PLO-3	PLO-4	PLO-5	PLO-6	PLO-7
CLO-1 สามารถออกแบบซอฟต์แวร์และระบบสารสนเทศที่ตอบสนองความต้องการของผู้ใช้งานได้เชิงมหภาค	<input type="checkbox"/>						
CLO-2 สามารถพัฒนาและทดสอบเครื่องเรียนเชิงเครื่องที่ตอบสนองความต้องการและทดสอบการทำงานได้	<input type="checkbox"/>						
CLO-3 สามารถออกแบบเครือข่ายคอมพิวเตอร์ที่ตอบสนองของผู้ใช้งานได้ตามมาตรฐาน	<input type="checkbox"/>						

Unsaved Changes | Save Part 7

รูปที่ ก.19: TQF3 Part 7 Curriculum Mapping

รูปที่ ก.20: TQF3 Part 7 Curriculum Mapping(Empty State)

### ก.1.5 ข้อจำกัดและเงื่อนไขการใช้งาน

ในการใช้งานระบบ ScoreOBE+ ผู้สอนต้องปฏิบัติตามข้อกำหนดและเงื่อนไขดังต่อไปนี้:

- ผู้สอนต้องกรอกข้อมูลใน มคอ.3 (TQF3) ให้เสร็จสมบูรณ์ก่อน จึงจะสามารถดำเนินการแก้ไข มคอ.5 (TQF5) ได้
- ผู้สอนต้องรับผิดชอบในการตรวจสอบความถูกต้องของคะแนนก่อนประกาศ เนื่องจากข้อมูลที่เผยแพร่ไปแล้วอาจมีผลกระทบต่อการประเมินผลการเรียนของนักศึกษา
- ผู้สอนสามารถกำหนดสิทธิ์ให้ผู้สอนร่วมเข้าถึงข้อมูลของบางกลุ่มเรียน แต่ ScoreOBE+ ไม่สามารถมอบสิทธิ์ให้ Co-Instructor มีอำนาจในการลบกระบวนการวิชาหรือแก้ไขกระบวนการวิชาได้

## ก.2 ผู้ดูแลหลักสูตร (Curriculum Administrator)

ผู้ดูแลหลักสูตรมีบทบาทสำคัญในการบริหารจัดการและติดตามเอกสารผลการเรียนรู้ของกระบวนการวิชา ต่าง ๆ ที่อยู่ภายใต้หลักสูตรที่ตนเองรับผิดชอบ โดยมีสิทธิ์และหน้าที่หลักดังต่อไปนี้

### ก.2.1 การจัดการเอกสาร TQF3 และ TQF5

ผู้ดูแลหลักสูตรสามารถเข้าถึงและแก้ไขเอกสาร **TQF3** (รายละเอียดกระบวนการวิชา) และ **TQF5** (รายงานผลการสอน) ได้สำหรับทุกกระบวนการวิชาที่อยู่ภายใต้หลักสูตรที่ตนมีสิทธิ์เข้าถึง โดยครอบคลุมทุกเลขอุปกรณ์ของกระบวนการวิชานั้น อย่างไรก็ตาม ผู้ดูแลหลักสูตรจะไม่สามารถเข้าถึงข้อมูลคะแนนของนักศึกษาในกระบวนการวิชาต่าง ๆ ได้ หากตนเองไม่ได้เป็นเจ้าของกระบวนการวิชา (Instructor) หรือผู้สอนร่วม (Co-Instructor) ในกระบวนการวิชานั้น ๆ

Section	Score 0	Score 1	Score 2	Score 3	Score 4	Number of Student	Average
001	2	0	0	0	25	27	3.70
002	0	0	0	0	28	28	4.00
003	0	0	0	2	26	28	3.93
004	0	0	0	0	28	28	4.00
005	2	0	0	6	20	28	3.50
006	0	0	0	0	28	28	4.00
801	1	0	0	0	11	12	3.67
802	1	0	0	0	10	11	3.64

รูปที่ ก.21: การจัดการเอกสาร TQF5

## ก.2.2 การติดตามสถานะเอกสาร TQF3 และ TQF5

ผู้ดูแลหลักสูตรสามารถตรวจสอบภาพรวมของสถานะเอกสาร **TQF3** และ **TQF5** ของทุกระบวนวิชา ในหลักสูตรของตนเองผ่านหน้าจอสีสุปพล โดยสามารถดูได้ว่ากระบวนวิชาใดได้ดำเนินการจัดทำเอกสารเสร็จสมบูรณ์แล้ว หรือกระบวนวิชาใดยังไม่ได้รับการดำเนินการ

Course No.	Course Name	Instructor	TQF 3	TQF 5	Action
000999	test	Thanaporn C.	No Data	No Data	
255217	Production Supporting Systems in Factories	Nirand R.	Done	Done	
259000	Innovative changemaking for sustainability	Thanatip C.	Done	No Data	
259192	Skills for Professionalism and Entrepreneurship	Thanatip C.	Done	Done	
259201	Computer Programming for Engineers	Done P.	No Data	No Data	
261102	Computer Programming	Karn P. Kasemsit T.	No Data	No Data	
261111	Internet and Online Community in the Age of AI	Thanatip C.	Done	No Data	
261112	Game Appreciation	Sakaguti R.	No Data	No Data	
261200	Object-Oriented Programming	Chinawat I. Navanon K.	No Data	No Data	

รูปที่ ก.22: การติดตามสถานะเอกสาร TQF3 และ TQF5

## ก.2.3 การจัดการกระบวนวิชา

ผู้ดูแลหลักสูตรสามารถบริหารจัดการและตรวจสอบกระบวนวิชาทั้งหมดที่อยู่ภายใต้หลักสูตรที่ตนมีสิทธิ์เข้าถึง โดยสามารถดูรายละเอียดของแต่ละกระบวนวิชาได้ เช่น ภาคการศึกษาที่เปิดสอน รายชื่อผู้สอนหลัก และผู้สอนร่วม รวมถึงเลขคุณภาพของกระบวนวิชานั้น ๆ หากมีการเปลี่ยนแปลงข้อมูล เช่น การเปลี่ยนแปลงรายชื่อผู้สอนหรือการเพิ่มกลุ่มเรียนใหม่ ผู้ดูแลหลักสูตรสามารถแก้ไขข้อมูล โดยการเปลี่ยนแปลงดังกล่าวจะมีผลต่อการเข้าถึงข้อมูลของผู้สอนและผู้สอนร่วมในกระบวนวิชานั้น ๆ

Course Management				
All Courses	CPE-2563	ME-2563	ME-INTER-2566	INB-2565
000999 test	Section 001	Active	Thanaporn C.	Semester 2
	Section 801	Active	Thanaporn C.	Semester 2
255217 Production Supporting Systems in Factories	Section 001	E-2563 Active	Nirand P.	Semester 2
	Section 002	E-2563 Active	Nirand P.	Semester 2
	Section 003	E-2563 Active	Nirand P.	Semester 2
	Section 004	E-2563 Active	Nirand P.	Semester 2
	Section 005	E-2563 Active	Nirand P.	Semester 2

รูปที่ ก.23: การจัดการกระบวนการวิชา

#### ก.2.4 การสร้างชุด PLO

ผู้ดูแลหลักสูตรสามารถสร้างชุด PLO ใหม่และกำหนดรายละเอียดเกี่ยวกับชุด PLO เช่น เกณฑ์ของ PLO จำนวนข้อ PLO และรายละเอียดของ PLO แต่ละข้อ อย่างไรก็ตาม หากต้องการนำชุด PLO ที่ตนเอง สร้างไปใช้ร่วมกับหลักสูตร จะต้องติดต่อผู้ดูแลระบบ

รูปที่ ก.24: การสร้างชุด PLO

#### ก.2.5 การเชื่อมโยง PLO Require กับ CLO ในแต่ละกระบวนการวิชา

ผู้ดูแลหลักสูตรสามารถกำหนดและเชื่อมโยง **PLO Require** (ผลลัพธ์การเรียนรู้ของหลักสูตรที่จำเป็น) กับ **CLO** (ผลลัพธ์การเรียนรู้ของกระบวนการวิชา) สำหรับแต่ละกระบวนการวิชาภายใต้หลักสูตรที่ตนมีสิทธิเข้าถึง โดยสามารถระบุได้ว่า PLO ใดมีความสำคัญและจำเป็นต้องได้รับการสนับสนุนจาก CLO ของกระบวนการวิชานั้น

นอกจากนี้ ระบบยังช่วยให้ผู้ดูแลหลักสูตรสามารถปรับปรุงและปรับแต่งการเชื่อมโยงระหว่าง CLO และ PLO ได้อย่างต่อเนื่อง

X PLO Management

Dashboard PLO 1/67 PLO 1/68 PLO ISNE 1/67

PLO 1/67 ⓘ  
unfulfilled ABET / ABET Criteria

PLO Description PLO Mapping

PLO Mapping 37 Courses

Cancel Save Changes

Course No.	PLO-1	PLO-2	PLO-3	PLO-4	PLO-5	PLO-6	PLO-7
259000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
261102	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
261103	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
261200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
261207	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
261208	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
261210	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
261212	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
261214	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

รูปที่ ก.25: การเชื่อมโยง PLO Require กับ CLO ในแต่ละกระบวนการวิชา

## ก.2.6 การวิเคราะห์การเชื่อมโยง CLO กับ PLO

ผู้ดูแลหลักสูตรสามารถดูการเชื่อมโยงระหว่าง **CLO (Course Learning Outcome)** กับ **PLO (Program Learning Outcome)** ในแต่ละกระบวนการวิชาภายใต้หลักสูตรที่รับผิดชอบ โดยสามารถตรวจสอบได้ว่า CLO ใดของกระบวนการวิชา มีการเชื่อมโยงกับ PLO ใดบ้าง เพื่อให้แน่ใจว่าการเรียนการสอนในแต่ละกระบวนการวิชาสอดคล้องกับผลลัพธ์การเรียนรู้ระดับหลักสูตร

ScoreOBE+

Welcome to ScoreOBE+  
Your courses are waiting on the right to jump in!  
Account? Top right menu

Course

Menu

TQF

CLO

PLO

CLO 2/67

Hi there, Worapitcha  
In CPE-2563 Curriculum have 24 Courses on this semester.

CPE-2563 ME-2563 ME-INTER-2566 INE-2565 EE-2567 EE-INTER-2567 CE-2566 CE-INTER-2567 ISNE-2566 ENV-2563 REAI-2564 MN-2567 IE-2563 IE-INTER-256

Course	CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7
259000 Innovative changemaking for sustainability	CLO 1 ⓘ	-	<input checked="" type="checkbox"/>	-	-	-	-	-
259000 Innovative changemaking for sustainability	CLO 2 ⓘ	-	-	<input checked="" type="checkbox"/>	-	-	<input checked="" type="checkbox"/>	-
259000 Innovative changemaking for sustainability	CLO 3 ⓘ	-	-	<input checked="" type="checkbox"/>	-	-	<input checked="" type="checkbox"/>	-
259000 Innovative changemaking for sustainability	CLO 4 ⓘ	-	<input checked="" type="checkbox"/>	-	<input checked="" type="checkbox"/>	-	-	-
261200 Object-Oriented Programming	TQF 3 No Data	-	-	-	-	-	-	-
261214 Microprocessor and Interfacing	TQF 3 No Data	-	-	-	-	-	-	-
261215 Embedded System Laboratory	TQF 3 No Data	-	-	-	-	-	-	-
261218 Algorithms for Computer Engineers	TQF 3 No Data	-	-	-	-	-	-	-
PLO Require: PLO 1, PLO 6, PLO 7								
261305 Operating Systems	TQF 3 No Data	-	-	-	-	-	-	-

รูปที่ ก.26: การวิเคราะห์การเชื่อมโยง CLO กับ PLO

## ก.2.7 การติดตามผลการเรียนรู้ของ PLO

ผู้ดูแลหลักสูตรสามารถดูข้อมูลสรุปเกี่ยวกับ **PLO** ในแต่ละกระบวนการวิชา ได้ โดยสามารถเข้าถึงข้อมูล คะแนนของ **PLO** ที่คำนวณจากผลการเรียนของนักศึกษาในกระบวนการวิชานั้นหลังจากที่มีการจัดทำเอกสาร **TQF5** เสร็จเรียบร้อยแล้ว นอกจากรายที่ ยังสามารถเลือกดูข้อมูลได้ในหลายมิติ ได้แก่

- รายเทอมการศึกษา – ดูผลการเรียนรู้ของ **PLO** สำหรับกระบวนการวิชาในแต่ละภาคการศึกษา

Course no.	Name	PLO-1	PLO-2	PLO-3	PLO-4	PLO-5	PLO-6	PLO-7
261494	Selected Topics in Computer Engineering (IT Infra and Cloud Tech.)	-	-	-	-	-	-	-
	Selected Topics in Computer Engineering (Blockchain Programming)	-	-	-	-	-	-	-
	Selected Topics in Computer Engineering (R Programming)	-	-	-	-	-	-	-
	Selected Topics in Computer Engineering (Startup Read)	-	-	-	-	-	-	-
	Selected Topics in Computer Engineering (Up Skills Unlock Limits)	-	-	-	-	-	-	-
261497	Selected Topics in Computer Software (The Cult of Privacy Technologies)	-	-	-	-	-	-	-
	Selected Topics in Computer Software (Full Stack Development)	-	-	-	-	-	-	-
261999	Course CPE Mock PLO Require: PLO 2, PLO 3, PLO 4, PLO 6, PLO 7	-	1.94	2.13	2.00	-	1.98	2.13

รูปที่ ก.27: การติดตามผลการเรียนรู้ของ PLO รายเทอมการศึกษา

- รายปีการศึกษา – ดูแนวโน้มผลลัพธ์การเรียนรู้ของ PLO ในกระบวนการวิชาเดี่ยวกันหรือหลาย กระบวนการวิชาตลอดทั้งปี

Year View		2567	Export PLO										
CPE-2563	ME-2563	ME-INTER-2566	INE-2565	EE-2567	EE-INTER-2567	CE-2566	CE-INTER-2567	ISNE-2566	ENV-2563	REAI-2564	MN-2567	IE-2563	IE-INTER-2566
Semester 1	0 Course	No Course Found											
Semester 2	3 Courses	261218 Algorithms for Computer Engineers	PLO 1 N/A PLO 6 N/A PLO 7 N/A	261361 Software Engineering	PLO 3 N/A PLO 4 N/A PLO 6 N/A	Semester 3	0 Course	No Course Found					
		261999 Course CPE Mock	PLO 2 1.94 PLO 3 2.13 PLO 4 2.00 PLO 6 1.98 PLO 7 2.13										

รูปที่ ก.28: การติดตามผลการเรียนรู้ของ PLO รายปีการศึกษา

- รายหลักสูตร – ดูภาพรวมผลการเรียนรู้ของ PLO ในทุกกระบวนการวิชาภายใต้หลักสูตรที่รับผิดชอบ

The screenshot shows the 'Curriculum View' interface. On the left, 'Step 1: Select Curriculum' lists 'CPE-2563 Curriculum' (Algorithms for Computer Engineers). On the right, 'Step 2: Select Courses' lists three courses: 261218 (Algorithms for Computer Engineers), 261361 (Software Engineering), and 261999 (Course CPE Mock). Below these, the 'CPE-2563 Curriculum' table shows PLOs 01-07 with their respective credit hours (N/A) and section numbers (501, 502, 503, 504, 505, 506, 507). A 'Clear Filter' button is at the bottom.

รูปที่ ก.29: การติดตามผลการเรียนรู้ของ PLO รายหลักสูตร

### ก.2.8 การนำออกคณ PLO และการจัดทำเอกสาร TQF7

ผู้ดูแลหลักสูตรสามารถนำออกคณ PLO เป็นไฟล์ .xlsx เพื่อนำไปใช้ในการจัดทำเอกสาร **TQF7** (รายงานผลการดำเนินการหลักสูตร) โดยสามารถเลือกได้ว่าต้องการนำเข้าข้อมูลจากกระบวนวิชาใดบ้าง เพื่อให้สามารถสรุปผลการเรียนรู้ของหลักสูตรในภาพรวมได้อย่างถูกต้องและครบถ้วน

The screenshot shows the 'Year View' interface for the academic year 2563. It displays three semesters: Semester 1 (0 Courses), Semester 2 (3 Courses), and Semester 3 (0 Courses). A modal window titled 'Export PLO' is open, showing a list of courses to export: 261218, 261361, and 261999. Below the modal, the 'CPE-2563 Curriculum' table for 261999 (Course CPE Mock) shows PLOs 02-07 with their respective credit hours (1.94, 2.13, 2.00, 1.98, 2.13). A 'Cancel' button and an 'Export PLO' button are at the bottom of the modal.

รูปที่ ก.30: การนำออกคณ PLO

### ก.2.9 ข้อจำกัดและเงื่อนไขการใช้งาน

- ผู้ดูแลหลักสูตรสามารถเข้าถึงเฉพาะกระบวนวิชาที่อยู่ในหลักสูตรที่ตนมีสิทธิ์เข้าถึงเท่านั้น ไม่สามารถเข้าถึงหลักสูตรอื่นที่ตนไม่มีสิทธิ์
- ผู้ดูแลหลักสูตรไม่สามารถเข้าถึงข้อมูลคณของนักศึกษาในแต่ละกระบวนวิชาได้ หากตนเองเมื่อได้เป็นเจ้าของหรือผู้สอนร่วมในกระบวนวิชานั้น
- การเขื่อมโยง PLO ที่ผู้ดูแลระบบสร้าง หากต้องการเขื่อมโยงกับหลักสูตรใด ๆ ต้องติดต่อผู้ดูแลระบบเพื่อเขื่อมโยงกับหลักสูตร

### ก.3 ผู้ดูแลระบบ (System Administrator)

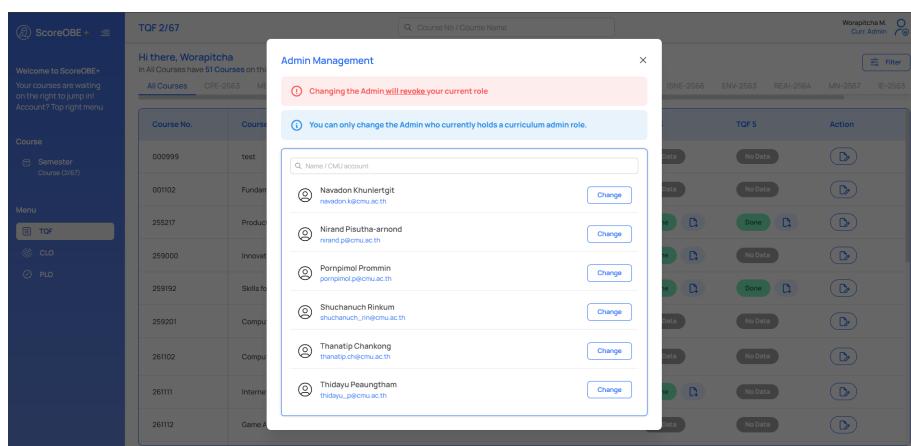
ผู้ดูแลระบบมีบทบาทสำคัญในการบริหารจัดการระบบ ScoreOBE+ เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพและรองรับการใช้งานของผู้ดูแลหลักสูตร อาจารย์ผู้สอน และนักศึกษา โดยมีสิทธิ์และหน้าที่หลักดังต่อไปนี้

#### ก.3.1 สิทธิ์การเข้าถึงของผู้ดูแลระบบ

เนื่องจากผู้ดูแลระบบเป็นบทบาทที่มีสิทธิ์มากกว่าผู้ดูแลหลักสูตร ผู้ดูแลระบบจึงมีสิทธิ์ในการเข้าถึงข้อมูลและการจัดการระบบเทียบเท่าหรือมากกว่าผู้ดูแลหลักสูตร ดังนั้น ผู้ดูแลระบบจะสามารถเข้าถึงฟังก์ชันการทำงานที่กล่าวไปข้างต้นได้ทั้งหมด

#### ก.3.2 การกำหนดผู้ดูแลระบบ

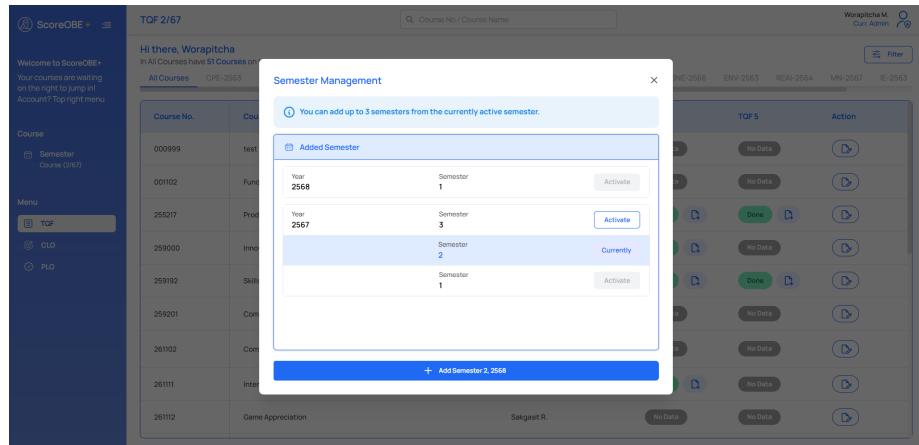
ผู้ดูแลระบบสามารถมีได้เพียงหนึ่งคนเท่านั้น หากต้องการเปลี่ยนแปลงผู้ดูแลระบบ ผู้ดูแลระบบคนปัจจุบันต้องเป็นผู้อนุมัติการเปลี่ยนแปลง โดยทำการพิมพ์ชื่อของบุคคลที่ต้องการแต่งตั้งเป็นผู้ดูแลระบบคนใหม่ ซึ่งบุคคลนั้นต้องเป็นผู้ที่อยู่ในบทบาทของผู้ดูแลหลักสูตรก่อนเท่านั้น



รูปที่ ก.31: การกำหนดผู้ดูแลระบบ

#### ก.3.3 การจัดการภาคการศึกษา

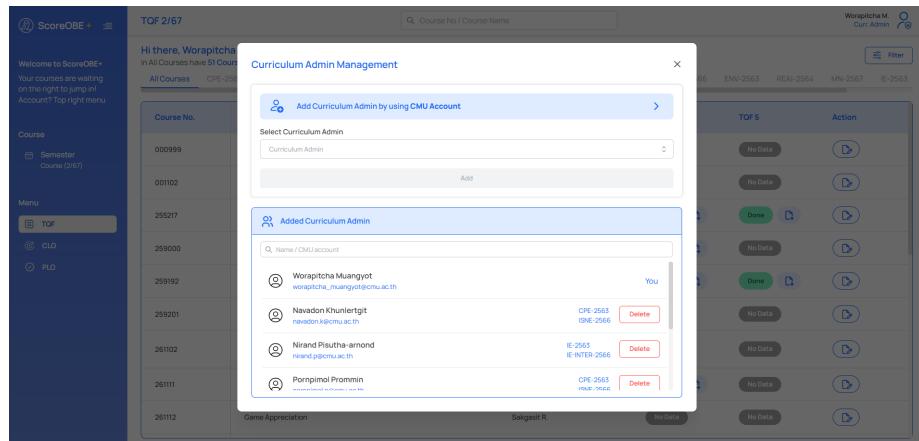
ผู้ดูแลระบบสามารถเพิ่มภาคการศึกษาใหม่และทำการเปิดใช้งาน (Activate) ภาคการศึกษานั้นได้ เมื่อมีการเปลี่ยนแปลงภาคการศึกษา ระบบจะใช้ค่าภาคการศึกษาที่เปิดใช้งานเป็นค่าเริ่มต้นในการจัดเก็บข้อมูลที่เกี่ยวข้องกับการเรียนการสอน



รูปที่ ก.32: การจัดการภาคการศึกษา

### ก.3.4 การจัดการผู้ดูแลหลักสูตร

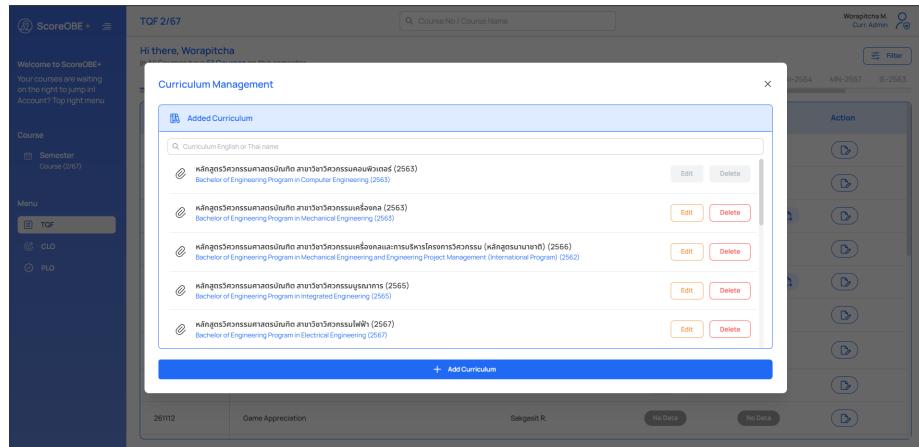
ผู้ดูแลระบบสามารถเพิ่มผู้ดูแลหลักสูตรได้ โดยสามารถเลือกบุคคลจากรายชื่อใน Dropdown หรือเพิ่มโดยใช้ CMU Account หลังจากเลือกบุคคลที่ต้องการเพิ่มแล้ว ผู้ดูแลระบบต้องกำหนดสิทธิ์การเข้าถึงหลักสูตรที่เกี่ยวข้องให้กับบุคคลดังกล่าว



รูปที่ ก.33: การจัดการผู้ดูแลหลักสูตร

### ก.3.5 การจัดการหลักสูตร

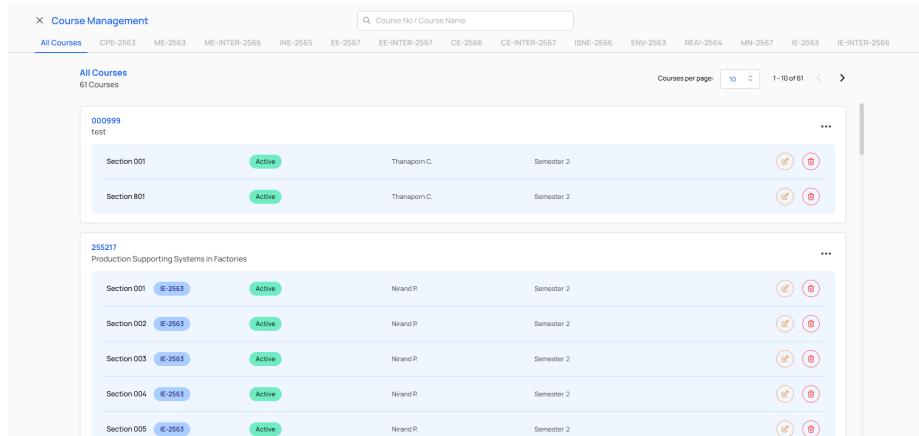
ผู้ดูแลระบบสามารถเพิ่มหลักสูตรใหม่ให้กับคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ได้ เมื่อมีหลักสูตรใหม่ ผู้ดูแลระบบสามารถเชื่อมโยงหลักสูตรนั้นเข้ากับชุด **PLO (Program Learning Outcome)** ที่มีอยู่หรือสร้างใหม่ได้



รูปที่ ก.34: การจัดการหลักสูตร

### ก.3.6 การจัดการกระบวนการวิชา

ผู้ดูแลระบบสามารถเข้าถึง และจัดการข้อมูลของทุกกระบวนการวิชาในทุกหลักสูตรภายใต้คณบดี ผู้ดูแลระบบ ไม่สามารถเข้าถึงข้อมูลใดๆ ของนักศึกษาได้ เว้นแต่ตนเองจะเป็นเจ้าของกระบวนการวิชา (Instructor) หรือผู้สอนร่วม (Co-Instructor) ในกระบวนการวิชานั้น



รูปที่ ก.35: การจัดการกระบวนการวิชา

### ก.3.7 ข้อจำกัดและเงื่อนไขการใช้งาน

- การเพิ่มภาคการศึกษาใหม่จะส่งผลต่อการบันทึกข้อมูลของระบบ จึงต้องตรวจสอบให้แน่ใจก่อนเปิดใช้งาน เมื่อเปิดใช้งานแล้ว ผู้สอนและผู้ดูแลหลักสูตรจะไม่สามารถแก้ไขข้อมูลใดๆ ในภาคการศึกษา ก่อนได้อีก และไม่สามารถย้อนกลับไปเปิดใช้งานภาคการศึกษาเดิมได้
- การปรับเปลี่ยนผู้ดูแลระบบคนใหม่ต้องได้รับการอนุมัติจากผู้ดูแลระบบคนปัจจุบัน หากพิมพ์ชื่อบุคคลต้องการแต่งตั้งเป็นผู้ดูแลระบบคนใหม่ไม่ถูกต้อง จะไม่สามารถเปลี่ยนแปลงได้

- ผู้ดูแลระบบไม่สามารถเข้าถึงหรือแก้ไขข้อมูลส่วนตัวของผู้ใช้งานคนอื่น ๆ ยกเว้นข้อมูลที่เกี่ยวข้องกับการจัดการ ScoreOBE+

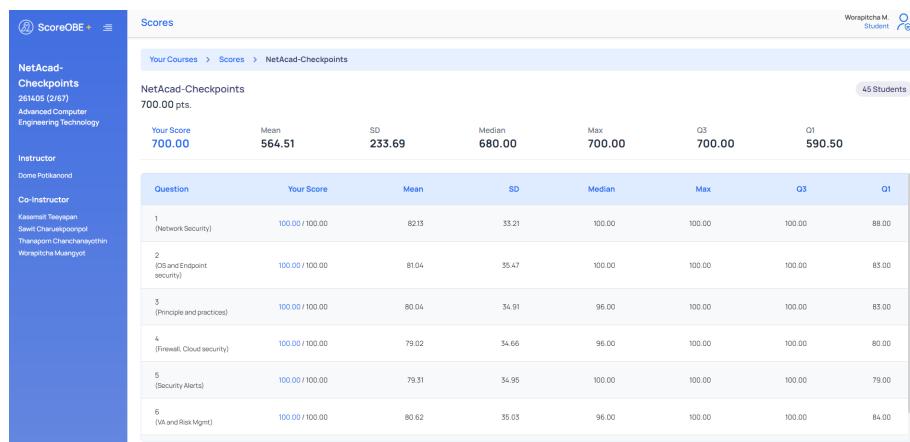
#### ก.4 นักศึกษา (Student)

นักศึกษาสามารถเข้าถึงข้อมูลผลการเรียนรู้ของตนเอง รวมถึงรายละเอียดที่เกี่ยวข้องกับการเรียนการสอนในระบบ ScoreOBE+ โดยมีสิทธิ์และความสามารถในการเข้าถึงข้อมูลต่าง ๆ ดังต่อไปนี้

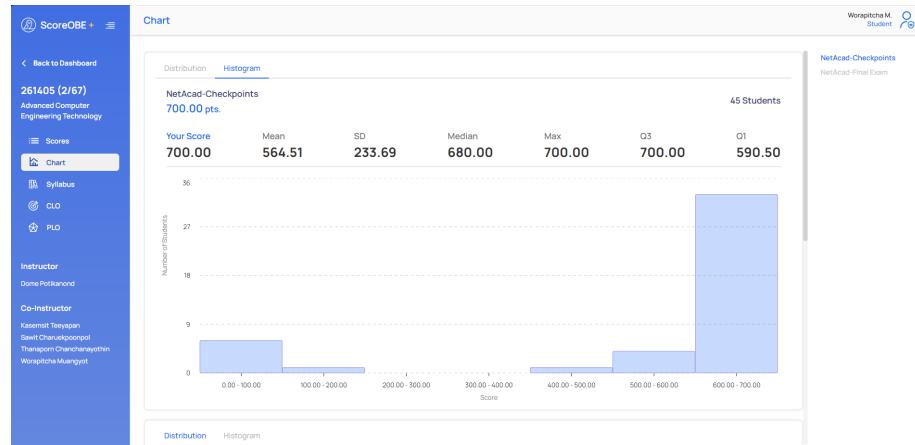
##### ก.4.1 การดูคะแนนและการวิเคราะห์ทางสถิติ

นักศึกษาสามารถเข้าถึงคะแนนของตนเองแต่ละกระบวนวิชาที่ลงทะเบียนเรียนได้ โดยสามารถดูคะแนนแยกตามองค์ประกอบของการประเมิน เช่น คะแนนสอบ คะแนนการบ้าน คะแนนโครงการ หรือองค์ประกอบอื่น ๆ ที่อาจารย์กำหนด นอกจากนี้ นักศึกษายังสามารถดูข้อมูลทางสถิติที่เกี่ยวข้อง ได้แก่

- ค่าเฉลี่ย ( $\mu$ ) ของคะแนนในกระบวนวิชา
- ค่ามัธยฐาน (Median)
- ค่าเบี่ยงเบนมาตรฐาน (Standard deviation)
- ค่าควอไทล์ (Quartile)
- ค่ามากที่สุด (Max)
- อิสโทแกรมแสดงการกระจายคะแนน เพื่อให้นักศึกษาเห็นภาพรวมของคะแนนของตนเองเบริ่ยบ เทียบกับเพื่อนร่วมชั้น



รูปที่ ก.36: ค่าทางสถิติ



รูปที่ ก.37: histogram ของคะแนน



รูปที่ ก.38: กราฟแสดงแนวโน้มการกระจายคะแนน

#### ก.4.2 การดูผลการเรียนรู้ของกระบวนวิชา (Course Learning Outcome, CLO)

นักศึกษาสามารถดูข้อมูลเกี่ยวกับ **Course Learning Outcome (CLO)** ของแต่ละกระบวนวิชาหลังจากที่อาจารย์ดำเนินการส่งเอกสาร **TQF3** แล้ว โดย CLO จะช่วยให้นักศึกษาเข้าใจวัตถุประสงค์ของรายวิชาและสิ่งที่คาดหวังให้ผู้เรียนบรรลุผลหลังจากการเรียนจบกระบวนวิชานั้น

ผลลัพธ์การเรียนรู้ของกระบวนวิชา  
Course Learning Outcome

**What is CLO?**

Course Learning Outcome (CLO) refers to what students should be able to know, understand, and do after completing a course. It serves as a guideline for both instructors and students to ensure that learning objectives are met effectively. CLOs are typically designed to be measurable and specific, focusing on different aspects of learning, such as: **Knowledge** (understanding theories, concepts, and principles), **Skills** (applying knowledge to solve problems or perform tasks), **Attitudes and Values** (developing ethical perspectives, teamwork, or communication skills).

CLO No.	CLO Description	Score	Evaluation
1	สามารถออกแบบซอฟต์แวร์และระบบสารสนเทศที่ตอบสนองความต้องการของผู้มีส่วนได้ส่วนเสียได้อย่างเหมาะสม Able to design a software and information system according to stakeholders requirements.	<span style="color: blue;">-</span>	<span style="color: grey;">-</span>
2	สามารถพัฒนาแบบจำลองด้านการเรียนรู้ของเครื่องที่ตอบสนองความต้องการและทดสอบการทํางานได้ Able to develop and to test a machine learning model according to requirements.	<span style="color: blue;">-</span>	<span style="color: grey;">-</span>
3	สามารถออกแบบเครือข่ายคอมพิวเตอร์ของจัดการศึกษาที่ตอบสนองความต้องการของผู้มีส่วนได้ส่วนเสียได้อย่างเหมาะสม Able to design a campus network according to stakeholders requirements.	<span style="color: blue;">-</span>	<span style="color: grey;">-</span>
4	สามารถออกแบบโครงสร้างพื้นฐานด้านศูนย์ข้อมูลที่ตอบสนองความต้องการของผู้มีส่วนได้ส่วนเสียได้อย่างเหมาะสม Able to design datacenter infrastructure according to stakeholders requirements.	<span style="color: blue;">-</span>	<span style="color: grey;">-</span>
5	สามารถออกแบบเชิงระบบคลาวด์การจัดการที่ออกแบบให้ถูกต้องและเหมาะสม Able to design how to use public cloud infrastructure to deploy software systems properly.	<span style="color: blue;">-</span>	<span style="color: grey;">-</span>
6	สามารถทำงานเป็นทีมโดยใช้เทคนิคการจัดการโครงการที่เหมาะสม Able to work as a team by utilizing appropriate project management techniques	<span style="color: blue;">-</span>	<span style="color: grey;">-</span>

รูปที่ ก.39: ผลลัพธ์การเรียนรู้ในมุ่งมองนักศึกษา

#### ก.4.3 การค้นหาข้อมูลโครงการสร้างรายวิชา (Course Syllabus)

หากนักศึกษาต้องการค้นหาข้อมูลเพิ่มเติมเกี่ยวกับรายวิชา นักศึกษาสามารถใช้ฟังก์ชัน **Search Course Syllabus** เพื่อค้นหารายวิชาที่ต้องการทราบรายละเอียดเพิ่มเติม เช่น เนื้อหาการเรียนรู้ วิธีการประเมิน และโครงสร้างของกระบวนวิชาในแต่ละภาคการศึกษา

**ScoreOBE+ Course View**

**261405 - Advanced Computer Engineering Technology**

- บทบัญญัติเกี่ยวกับเทคโนโลยีคอมพิวเตอร์ที่มีสูง แพลตฟอร์มของเทคโนโลยีที่มีความทันสมัย เช่น โครงสร้างข้อมูลที่ใช้ในเทคโนโลยีคอมพิวเตอร์ที่มีสูง การประยุกต์ใช้เทคโนโลยีที่มีความทันสมัย เช่น กลไกหน่วยและอัลกอริズึมที่ใช้กับวิเคราะห์ความคิดพิเศษ
- Introduction to advanced computer engineering technologies. Platforms of advanced computer engineering technologies. Tools used in advanced computer engineering technologies. Applications of advanced computer engineering technologies. Laws and ethics related to computer engineering.

**Part 1 - ข้อมูลกระบวนวิชา Course Information**

หลักสูตร Curriculum	สาหรับหลักสูตร
ประเภทกระบวนวิชา Course Type	วิชาเฉพาะ (Field of Specialization)
ปีที่เรียน Student Year	ปีที่ 3 3rd year
	ปีที่ 4 4th year
	ปีที่ 5 5th year
	ปีที่ 6 6th year
ผู้สอนหลัก Main Instructor	ใบอนุญาตฯ

รูปที่ ก.40: รายละเอียดกระบวนวิชาที่นักศึกษาลงที่เบียนเรียน

รูปที่ ก.41: ค้นหารายละเอียดกระบวนการวิชา

#### ก.4.4 การดูผลการเรียนรู้ระดับหลักสูตร (Program Learning Outcome, PLO)

นอกจากการดู CLO แล้ว นักศึกษาสามารถดู **Program Learning Outcome (PLO)** ซึ่งเป็นผลการเรียนรู้ในระดับหลักสูตรที่เชื่อมโยงกับ CLO ของแต่ละกระบวนการวิชา ข้อมูลนี้จะแสดงผลในรูปแบบ **แผนภูมิแท่ง (Bar Chart)** เพื่อให้เข้าใจความก้าวหน้าและพัฒนาการของผลการเรียนรู้ในระดับหลักสูตรได้อย่างง่ายดาย

PLO	Score	Courses
PLO 1	0	0 courses
PLO 2	1.5	1 course
PLO 3	1.5	1 course
PLO 4	1.5	1 course
PLO 5	1.5	0 courses
PLO 6	1.5	1 course
PLO 7	1.5	1 course

รูปที่ ก.42: ผลลัพธ์การเรียนรู้ตลอดหลักสูตร

#### ก.4.5 ข้อจำกัดและเงื่อนไขการเข้าถึงข้อมูล

- นักศึกษาสามารถเข้าถึงข้อมูลคณะแนนของตนเองเท่านั้น ไม่สามารถเข้าถึงข้อมูลคณะแนนของนักศึกษาคนอื่นได้
- การดู CLO และ PLO จะเป็นไปตามข้อมูลที่อาจารย์หรือผู้ดูแลหลักสูตรอัปเดตในระบบ หากข้อมูลยังไม่ถูกส่งหรืออัปเดต นักศึกษาจะไม่สามารถเข้าถึงข้อมูลดังกล่าวได้
- การเข้าถึง Course Syllabus สามารถทำได้ผ่านฟังก์ชันการค้นหา และสามารถดูเฉพาะกระบวนวิชาที่เปิดสอนในหลักสูตรที่นักศึกษาเกี่ยวข้อง
- ระบบอาจมีข้อจำกัดในการแสดงข้อมูลบางประเภทขึ้นอยู่กับสิทธิ์การเข้าถึงที่กำหนดโดยผู้ดูแลหลักสูตรและอาจารย์ผู้สอน