



ระบบจัดการอุปกรณ์อำนวยความสะดวกในห้องปฏิบัติการ AMDRL
AMDRL LAB's Equipment Management System

เนติธร ปรีชาพัฒนานนท์

โครงการฝึกงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยี ไทย-ญี่ปุ่น
พ.ศ. 2564

ระบบจัดการอุปกรณ์อำนวยความสะดวกในห้องปฏิบัติการ AMDRL
AMDRL LAB's Equipment Management System

เนติธร ปรีชาพัฒนานนท์

โครงการฝึกงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยี ไทย-ญี่ปุ่น
พ.ศ. 2564

คณะกรรมการสอบ

..... อาจารย์ที่ปรึกษา ประธานกรรมการสอบ
(ดร.ไพศาล สุตวิสัย)

..... กรรมการสอบ
(รศ.ดร.ไตรสิทธิ์ เบญจบุญยสิทธิ์)

..... กรรมการสอบ
(ดร.ชาตรี บุชบาธ)

..... ประธานสหกิจศึกษาสาขาวิชา กรรมการสอบ
(ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ธรรมตระกูล)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย – ญี่ปุ่น

ชื่อโครงการ	ระบบจัดการอุปกรณ์อำนวยความสะดวกในห้องปฏิบัติการ AMDRL
ผู้เขียน	นายเนติธร ปรีชาพัฒนานนท์
คณะวิชา	วิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา	ดร.ไพศาล สดวิสัย
พนักงานที่ปรึกษา	ดร.ไพศาล สดวิสัย
ชื่อบริษัท	สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ประเภทธุรกิจ/สินค้า	สถาบันอุดมศึกษาเอกชน

บทสรุป

ห้องปฏิบัติการนั้นเป็นสถานที่สำหรับการทำงานด้านการวิจัยโดยภายในห้องปฏิบัติการนั้นจะมีสิ่งอำนวยความสะดวกสำหรับผู้ทำงานวิจัย ซึ่งทรัพยากรสิ่งอำนวยความสะดวกเหล่านั้นผู้ทำงานวิจัยสามารถนำไปใช้ได้ตามความสะดวก

อย่างไรก็ตามถึงแม้ว่าทรัพยากรสิ่งอำนวยความสะดวกภายในห้องปฏิบัติการนั้นจะมีหลากหลายประเภทเพื่อรองรับการทำงานวิจัยในหลากหลายหัวข้อและรูปแบบ ปัญหาเรื่องจำนวนของผู้ทำงานวิจัยก็ยังส่งผลให้จำนวนอุปกรณ์อำนวยความสะดวกในห้องปฏิบัติการนั้นอาจไม่เพียงพอต่อความต้องการของผู้ทำงานวิจัยจึงเกิดการเวียนใช้อุปกรณ์อำนวยความสะดวกซึ่งการจะติดตามอุปกรณ์นั้นเป็นไปได้ยากหากไม่ทราบว่าผู้วิจัยที่ได้นำไปใช้ก่อนหน้านี้เป็นใคร

ดังนั้นการทำให้ระบบการติดตามอุปกรณ์จึงเป็นวิธีการช่วยผู้วิจัยในการติดตามและค้นหาอุปกรณ์ รวมถึงวิธีการที่จะเข้าถึงระบบการติดตามอุปกรณ์นั้นจำเป็นจะต้องเข้าถึงเพื่อติดตามได้ง่ายดังนั้นการพัฒนาการติดตามอุปกรณ์จึงพัฒนาขึ้นมาจากพื้นฐานของ web application ที่มี cloud database จึงจะทำให้สามารถเข้าถึงได้ง่ายจากหลากหลายอุปกรณ์ที่สามารถเข้าถึง internet ได้

Project's name	AMDRL LAB's Equipment Management System
Author	Mr.Netithorn Preechapattananont
Faculty	Engineering, Department of Electrical Engineering
Faculty Advisor	Dr.Phaisarn Sudwilai
Job Supervisor	Dr.Phaisarn Sudwilai
Company's name	Thai-Nichi Institute of Technology
Business Type / Product	Private Higher Education Institution

Summary

Laboratory is the research workplace where there are plenty of equipments and supporting tools for researchers. Those kind of equipments are prepared for the convenience and ease of use for the researchers.

However, with many types of such tools to support mutiple kinds of research works as well as the high number of researchers working in the same period affects to the insufficiency of researcher demands. Those equipments are tended to be reused. To track those tools is quite difficult as the previous users may be unknown.

Therefore, building this management system is considered as the solution to help the researchers/users for tracking and searching. In addition, accessibility to this system should be uncomplicated. Hence, this system was developed with the foundation of web application and cloud database to support the usage with mutiple devices with internet connection.

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณอาจารย์ ดร.ไพศาล สุตวิสัย ที่ปรึกษาโครงการฝึกงาน ซึ่งสละเวลาอันมีค่ายิ่งใหญ่คำแนะนำและแนะแนวทางการแก้ไขปัญหาที่เกิดขึ้นตลอดการดำเนินโครงการ

ข้าพเจ้าขอขอบพระคุณ ดร.ชาตรี บุษบาธร และผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ธรรมตระกูล ที่กรุณาเป็นคณะกรรมการสอบโครงการฝึกงานในครั้งนี้

เนติธร ปรีชาพัฒนานนท์

ผู้จัดทำ

สารบัญ

	หน้า
บทสรุป	ก
Summary	ข
กิตติกรรมประกาศ	ฅ
สารบัญ	ค
สารบัญตาราง	ฅ
สารบัญรูป	ฉ
บทที่ 1 บทนำ	
1.1 ชื่อและที่ตั้งของสถานประกอบการ	1
1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร	1
1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร	2
1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย	2
1.5 พนักงานที่ปรึกษา และ ตำแหน่งของพนักงานที่ปรึกษา	2
1.6 ระยะเวลาที่ปฏิบัติงาน	2
1.7 ที่มาและความสำคัญของปัญหา	3
1.8 วัตถุประสงค์หรือจุดมุ่งหมายของโครงการ	3
1.9 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย	3
1.10 นิยามศัพท์เฉพาะ	3

สารบัญ (ต่อ)

	หน้า
บทที่ 2 ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน	
2.1 ทฤษฎีที่เกี่ยวข้อง	4
บทที่ 3 แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน	
3.1 แผนการฝึกงาน	8
3.2 ขั้นตอนการดำเนินโครงการ	9
3.3 ขั้นตอนการพัฒนาโปรแกรมส่วนการประมวลผลฝั่งเซิร์ฟเวอร์ด้วย ภาษา Python และ Flask Framework	9
บทที่ 4 สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่างๆ	
4.1 ขั้นตอนและผลการดำเนินงาน	24
บทที่ 5 บทสรุปและข้อเสนอแนะ	
5.1 ทฤษฎีที่เกี่ยวข้อง	30
5.2 แนวทางการแก้ไขปัญหา	30
5.3 ข้อเสนอแนะจากการดำเนินงาน	30
เอกสารอ้างอิง	31
ประวัติผู้จัดทำโครงการ	33

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 3.1 แผนการฝึกงาน 1 กรกฎาคม 2564 – 1 สิงหาคม 2564	7

สารบัญรูป

รูปที่		หน้า
1.1	แผนผังโครงสร้างองค์กร	2
2.1	รูปแบบโครงสร้างของภาษา html	4
2.2	ตัวอย่างภาษา CSS ในโครงสร้างของภาษา html	5
2.3	ตัวอย่างภาษา Java Script ในโครงสร้างของภาษา html	5
3.1	ขั้นตอนการดำเนินงาน	9
3.2	รายการ Extension ที่จำเป็นในการเขียนสร้างระบบจัดการอุปกรณ์	10
3.3	การตั้งค่าการเชื่อมต่อกับ Cloud database และการเริ่มต้นการทำงานของ Extension ต่างๆ	10
3.4	โครงสร้างและข้อกำหนดของฐานข้อมูลอุปกรณ์ และ สมาชิก	11
3.5	ตัวอย่างข้อมูลอุปกรณ์ที่ถูกจัดเก็บอยู่บน Cloud database	11
3.6	ตัวอย่างข้อมูลสมาชิกที่ถูกจัดเก็บอยู่บน Cloud database	12
3.7	รูปแบบ Form ที่มีเงื่อนไขสำหรับเก็บข้อมูลของการ login sign-up และ อุปกรณ์	12
3.8	หน้าต่างของ index ของระบบจัดการอุปกรณ์	13
3.9	โค้ดการทำงานในฝั่งของเซิร์ฟเวอร์ในหน้าของ index และ login	13
3.10	หน้าต่าง login	14
3.11	โค้ด html ในส่วนของหน้าต่าง login	14
3.12	โค้ดการทำงานในฝั่งของเซิร์ฟเวอร์ในหน้าของ signup	15
3.13	โค้ด html ในส่วนของหน้าต่าง signup	15
3.14	หน้าต่าง signup	16
3.15	หน้าต่าง dashboard ของผู้ใช้งาน (สมาชิกธรรมดา)	16
3.16	โค้ดการทำงานในฝั่งของเซิร์ฟเวอร์ในหน้าของ dashboard ที่มีการดึงข้อมูลและแก้ไขข้อมูลสถานะไปยัง Cloud database	17
3.17	โค้ด html ในหน้าต่าง dashboard ในส่วนของการเลือกประเภทของข้อมูลที่ต้องการให้แสดง	18
3.18	โค้ด html ในหน้าต่าง dashboard ในส่วนของการแสดงผลข้อมูลหลังจากการดึงข้อมูลในฝั่งของเซิร์ฟเวอร์และการแก้ไขสถานะ ยืม/คืน ของอุปกรณ์	18
3.19	หน้าต่าง dashboard ของผู้ใช้งาน (admin)	19
3.20	หน้าต่าง dashboard สำหรับการจัดการอุปกรณ์	19

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.21 โค้ดการทำงานในฝั่งของเซิร์ฟเวอร์ในหน้าของ Manage devices ที่มีการดึงข้อมูล แก้ไขข้อมูลไปและ เพิ่มข้อมูลไปยัง Cloud database	20
3.22 หน้าต่าง pop-up form ในการเพิ่มอุปกรณ์	20
3.23 โค้ด html ในหน้าต่าง Manage devices ในส่วนของ pop-up form ที่เก็บข้อมูลของอุปกรณ์ ก่อนส่งข้อมูลไปต่อยังการประมวลผลฝั่งเซิร์ฟเวอร์เพื่อเพิ่มข้อมูล	21
3.24 หน้าต่าง pop-up form ในการแก้ไขข้อมูลของอุปกรณ์	21
3.25 โค้ด html ในหน้าต่าง Manage devices ในส่วนของ pop-up form ที่เก็บข้อมูลของอุปกรณ์ ก่อนส่งข้อมูลไปต่อยังการประมวลผลฝั่งเซิร์ฟเวอร์เพื่อแก้ไขข้อมูล	22
3.26 โค้ด html ในหน้าต่าง Manage devices ในส่วนของ pop-up form ที่เก็บข้อมูลของอุปกรณ์ ก่อนส่งข้อมูลไปต่อยังการประมวลผลฝั่งเซิร์ฟเวอร์เพื่อแก้ไขข้อมูล (ต่อ)	23
4.1 การ sign-up	24
4.2 ข้อมูลผู้ใช้ใน cloud database	24
4.3 การ login เข้าสู่ระบบ	25
4.4 หน้า dashboard สำหรับการใช้งาน ยืม/คืน อุปกรณ์	25
4.5 การเพิ่มอุปกรณ์ในกรณีที่เป็น admin	26
4.6 อุปกรณ์ถูกเพิ่ม	26
4.7 การแก้ไขข้อมูลอุปกรณ์	27
4.8 ข้อมูลอุปกรณ์ถูกแก้ไข	27
4.9 การลบอุปกรณ์ออกจากฐานข้อมูล	28
4.10 หัวข้อบน Nav-bar ในหน้า index	28
4.11 ส่วนประกอบในหน้า Dashboard	29
4.12 ส่วนประกอบในหน้า Manage devices	29

บทที่ 1

บทนำ

1.1 ชื่อและสถานที่ตั้งของสถานประกอบการ

ชื่อสถานประกอบการ ห้องปฏิบัติการ AMDRL (C-402)

ที่ตั้งสถานประกอบการ 1771/1 ซ.พัฒนาการ37 ถนนพัฒนาการ แขวงสวนหลวง

เขตสวนหลวง กรุงเทพฯ 10250

Tel. 0-2763-2600, Fax. 0-2763-2700

1.2 ลักษณะธุรกิจของสถานประกอบการ หรือ การให้บริการหลักขององค์กร

สถาบันเทคโนโลยีไทย-ญี่ปุ่น เป็นสถาบันอุดมศึกษาเอกชนชั้นนำทางด้านเทคโนโลยีและการจัดการ ที่เน้นทักษะการ สื่อสาร มีเครือข่ายทั้งในและต่างประเทศที่เข้มแข็ง มีความเป็นเลิศทางวิชาการและการ ประยุกต์ เป็นศูนย์กลาง เผยแพร่องค์ความรู้แก่สังคม โดย

(1) จัดการ การศึกษาระดับอุดมศึกษาในสาขาวิชาชีพเฉพาะทางขั้นสูงที่เป็นความ ต้องการ ของ ภาคธุรกิจและภาคอุตสาหกรรม โดยมุ่งเน้น ความเป็นเลิศทางวิชาการการปฏิบัติ และ การ ประยุกต์ใช้จริง

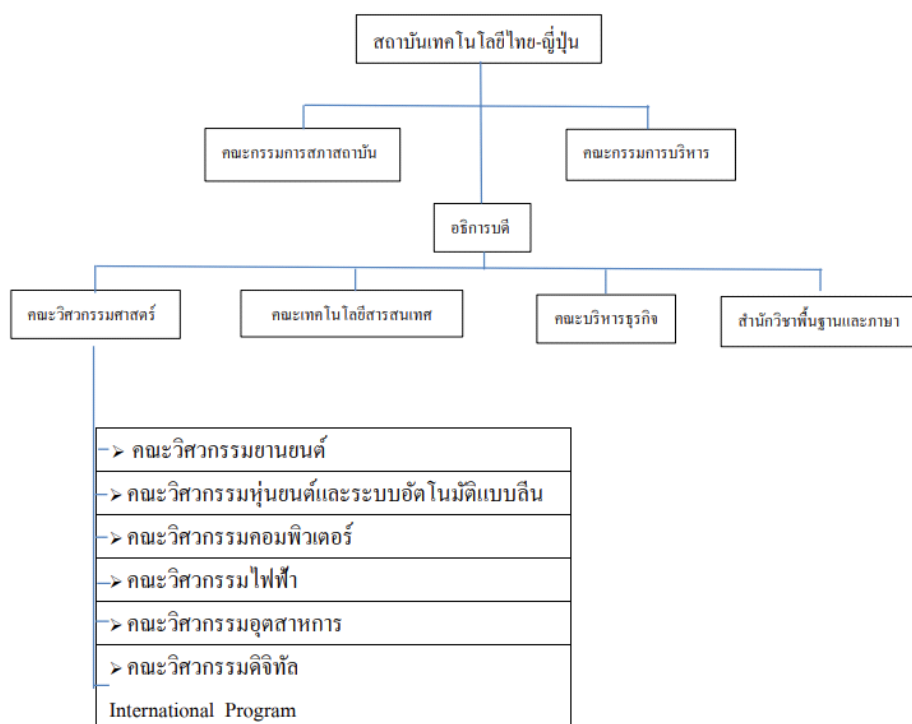
(2) พัฒนานักศึกษาให้มีความรู้คู่คุณธรรม คิดเป็น ทำเป็น มีความรับผิดชอบในการ ทำงาน เป็นแบบอย่างที่ดีและมีจิตสำนึกต่อสังคม

(3) ดำเนินการวิจัย สร้างสรรค์และพัฒนาเทคโนโลยีและองค์ความรู้ใหม่ๆ เพื่อ สนับสนุนการ เรียนการสอน การพัฒนาภาคธุรกิจและอุตสาหกรรม

(4) ถ่ายทอดความรู้และเทคโนโลยีขั้นสูง เพื่อเสริมสร้างความสามารถในการแข่งขันของภาค ธุรกิจและอุตสาหกรรม

(5) ทำนุบำรุง ส่งเสริม เผยแพร่และแลกเปลี่ยนศิลปวัฒนธรรมและภูมิปัญญาไทย

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร



รูปที่ 1.1 แผนผังโครงสร้างองค์กร

1.4 ตำแหน่งและหน้าที่การงานที่นักศึกษาได้รับ

ตำแหน่งนักศึกษาฝึกงาน แผนกวิศวกรรมไฟฟ้า

ศึกษาวิธีการเขียนโปรแกรมและการใช้ Framework ต่างๆเพื่อพัฒนา Web application
สำหรับการจัดการอุปกรณ์ในห้องปฏิบัติการ

1.5 พนักงานที่ปรึกษา และ ตำแหน่งของพนักงานที่ปรึกษา

อาจารย์ ดร.ไพศาล สุทธิวิทย์

1.6 ระยะเวลาปฏิบัติงาน

1 กรกฎาคม 2564 – 1 สิงหาคม 2564

1.7 ที่มาและความสำคัญของปัญหา

เนื่องด้วยทรัพยากรสิ่งอำนวยความสะดวกในห้องปฏิบัติการมีอยู่หลากหลายประเภทแต่มีอยู่อย่างจำกัดจึงทำให้เมื่อเกิดเหตุการณ์ที่มีผู้จำเป็นต้องการใช้งานแต่อุปกรณ์ที่ต้องการใช้นั้นไม่พอจึงจำเป็นต้องเกิดการเวียนกันใช้ขึ้นทำให้การติดตามอุปกรณ์ขึ้นนั้นเป็นเรื่องยากเนื่องจากสมาชิกคนไหนก็สามารถนำไปใช้ได้

1.8 ขอบเขตการศึกษา

1.8.1 ออกแบบและสร้างระบบการจัดการอุปกรณ์ภายในห้องปฏิบัติการในรูปแบบของ web application ที่เชื่อมต่อกับ cloud database

1.8.2 ออกแบบและสร้างฐานข้อมูลสำหรับจัดการอุปกรณ์ในห้องปฏิบัติการโดยระบุคุณสมบัติของอุปกรณ์และสถานะของอุปกรณ์

1.9 ประโยชน์ที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

1.9.1 ทักษะการเขียนโปรแกรมภาษา Python โดยใช้ Flask Framework

1.9.2 ทักษะการออกแบบและจัดการฐานข้อมูล

1.9.2 ทักษะการออกแบบเว็บไซต์เบื้องต้นเพื่อง่ายต่อการใช้งาน

1.10 นิยามศัพท์เฉพาะ

1.10.1 Web Application คือ Application ที่ถูกเขียนขึ้นมาให้สามารถเปิดใช้ใน Web browser ได้โดยตรงไม่จำเป็นต้องดาวโหลดและติดตั้งเพิ่มเติมเพื่อใช้งาน

1.10.2 Cloud Database คือ ระบบของฐานข้อมูลรูปแบบหนึ่งที่สามารถทำงานได้แบบออนไลน์ สามารถทำงานเชื่อมโยงได้กับ Web Application ได้อย่างสะดวกโดยฐานข้อมูลนี้ทำงานอยู่บนเทคโนโลยี Cloud ทำให้ฐานข้อมูลมีความยืดหยุ่นสามารถรับส่งข้อมูลได้อย่าง Realtime

1.10.3 Framework คือ โครงสร้างของโปรแกรมที่ถูกพัฒนาขึ้นเป็นพื้นฐานในการเขียนโปรแกรมเพื่ออำนวยความสะดวกโดยมีวัตถุประสงค์เฉพาะทาง

1.10.4 Extension คือ ส่วนขยายเพิ่มเติมในการเขียนโปรแกรมที่มี function สำเร็จรูปบางส่วนเพื่อช่วยในการเขียนโปรแกรมได้สะดวกขึ้น

1.10.5 Redirect คือ การเปลี่ยนเส้นทางของ URL จากเดิมที่แสดงหน้าเว็บหนึ่งเป็น URL ปลายทางเพื่อเปลี่ยนหน้าเว็บที่แสดงผล

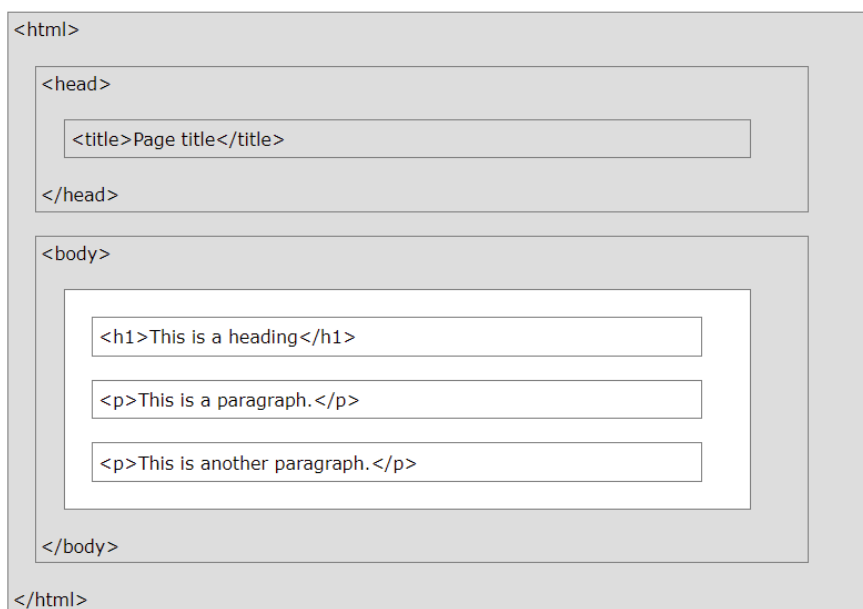
บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.1 ทฤษฎีที่เกี่ยวข้อง

องค์ประกอบของเว็บไซต์ [1]

เว็บไซต์นั้นสามารถแบ่งองค์ประกอบคร่าวๆได้สามส่วนโดยประกอบไปด้วยโครงสร้างหลักของเว็บไซต์การตกแต่งภายในเว็บไซต์และฟังก์ชันการมีปฏิสัมพันธ์กับผู้ใช้โดยโครงสร้างหลักของเว็บไซต์นั้นจะถูกพัฒนาขึ้นด้วยภาษา HTML (HyperText Markup Language) ซึ่งในส่วนนี้จะทำการขึ้นโครงสร้างหลักๆของเว็บไซต์เช่นการแบ่งส่วนการแสดงผลเนื้อหาส่วนหัวส่วนเนื้อหาของเว็บไซต์โดยใช้แท็กซึ่งระบุคุณสมบัติของเนื้อหาภายในแท็กนั้นและยังสามารถระบุส่วนเสริมที่จะนำมาเรียกใช้ได้โดยแท็กหลักๆที่ใช้เป็นโครงสร้างเว็บไซต์ประกอบไปด้วยแท็กหัว แท็กบอดี โดยนอกเหนือจากนี้จะมีแท็กย่อยๆที่นำไปใช้ออกแบบการจัดเรียงอีกด้วย



รูปที่ 2.1 รูปแบบโครงสร้างของภาษา html

ในส่วนของการตกแต่งภายในเว็บไซต์นั้นจะใช้ภาษา CSS (Cascading Style Sheet) โดยการจะนำมาใช้งานนั้นจำเป็นที่จะต้องประกาศระบุเป็นส่วนเสริมในแท็กหัวของ HTML ก่อนดังที่บอกใน

ข้างต้นหลังจากนั้นจึงนำไปเขียนฝังไว้ในแท็กต่างๆของ HTML เพื่อที่จะตกแต่งเนื้อหาในแท็กๆนั้นๆ ไม่ว่าจะเป็น สี ขนาด และ กำหนดระยะ เป็นต้น โดยยังสามารถสร้างคลาสเพื่อกำหนดรูปแบบการตกแต่งไว้ และนำไปใส่ในแท็กของ HTML เพื่อที่จะทำให้สะดวกต่อการตกแต่งมากขึ้นในกรณีที่มีเนื้อหาที่เราต้องการตกแต่งแบบเดียวกันหลายๆเนื้อหา

```
body {
    background-color: lightblue;
}

h1 {
    color: white;
    text-align: center;
}

p {
    font-family: verdana;
    font-size: 20px;
}
```

รูปที่ 2.2 ตัวอย่างภาษา CSS ในโครงสร้างของภาษา html

สุดท้ายในส่วนของการเพิ่มความสามารถให้หน้าเว็บไซต์และทำให้หน้าเว็บไซต์มีปฏิสัมพันธ์กับ ผู้ใช้นั้นจำเป็นที่จะต้องพัฒนาด้วยภาษา JavaScript โดยวิธีที่จะนำมาใช้นั้นจำเป็นต้องประกาศระบุเป็น ส่วนเสริมในแท็กหัวของ HTML ก่อนเช่นเดียวกับการเรียกใช้งาน CSS โดยการใช้งานก็เช่นเดียวกับ CSS คือการเขียนนำไปฝังไว้ในแท็กของ HTML หรือในส่วนของเนื้อหาเพื่อเพิ่มความสามารถให้กับเนื้อหาของ แท็กนั้นๆ

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

รูปที่ 2.3 ตัวอย่างภาษา Java Script ในโครงสร้างของภาษา html

นอกจากนี้ยังมี Framework ที่ชื่อว่า Bootstrap [2] [3] ซึ่งเป็น Framework นำในส่วนของ CSS และ JavaScript มารวมกันสำเร็จรูปพร้อมนำไปใช้งานโดยการเรียกใช้เป็นคลาสในการเพิ่มความสามารถให้กับหน้าเว็บไซต์โดยการใช้งานนั้นก็เป็นเช่นเดียวกับการใช้งาน CSS และ JavaScript ที่กล่าวไว้ข้างต้นโดยการนำไปใส่ไว้ในแท็กของ HTML ก่อนโดยการเพิ่มเข้าไปยัง HTML นั้นสามารถเพิ่มเข้าไปได้สองวิธีคือการเพิ่มแบบออนไลน์โดยการใส่ลิงค์ที่ทาง HTML นั้นจะไปเรียกใช้โดยวิธีนี้ในการใช้งานเว็บไซต์นั้นจำเป็นที่จะต้องเชื่อมต่ออินเทอร์เน็ตตลอดเวลาเพื่อที่จะให้หน้าเว็บสามารถเรียกใช้ Bootstrap จาก server ได้ ส่วนอีกวิธีหนึ่งคือการโหลด Bootstrap เอาไว้ในเครื่องที่เราจะใช้งานจากนั้นทำการใส่เส้นทางของ Bootstrap ภายในเครื่องไว้ในส่วนของการเรียกใช้ใน HTML โดยวิธีนี้จะทำให้เว็บสามารถดึง Bootstrap มาใช้ได้โดยไม่ต้องเชื่อมต่ออินเทอร์เน็ต

ฐานข้อมูล (Database) [4] [5] [7]

ฐานข้อมูลคือกลุ่มข้อมูลที่ได้ถูกเก็บไว้ในรูปแบบของตารางโดยมีโครงสร้างรูปแบบอย่างมีระบบตามที่ผู้ออกแบบสนใจและกำหนดไว้โดยภายในฐานข้อมูล ข้อมูลอาจจะมีความสัมพันธ์ซึ่งกันและกันโดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลายๆแฟ้มข้อมูล ซึ่งการจัดการข้อมูลในฐานข้อมูลนั้นจำเป็นต้องเขียนโปรแกรมมาจัดการโดยภาษาหนึ่งที่ยิยมใช้ในการจัดการข้อมูลนั้นคือ SQL (Structured Query Language) โดยภาษา SQL นั้นสามารถใช้งานร่วมกับเว็บไซต์ระบบฐานข้อมูล SQL Server ไปจนถึงการสร้างระบบวิเคราะห์ข้อมูลโดยเมื่อพูดถึงการจัดการข้อมูลขนาดใหญ่ระดับ Big Data แล้วการใช้โปรแกรมอย่าง Excel และ Spreadsheet นั้นจะมีประสิทธิภาพและความเสถียรน้อยกว่าในการจัดการข้อมูลเมื่อเทียบกับ SQL

ภาษา Python [6]

ภาษาโปรแกรม Python นั้นเป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูง (High-level programming language) โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เราเขียนให้เป็นภาษาเครื่อง Python มีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่เรต้องการ นอกจากนั้นภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language)

Flask Framework [8] [9]

Flask คือ Framework ที่ใช้ในการสร้าง Web Application ที่รองรับ Web Server Gateway Interface (WSGI) Flask เป็น Framework ที่ออกแบบให้มีขนาดเล็ก ไม่ซับซ้อน มีฟังก์ชันเท่าที่จำเป็นเท่านั้น ถึงขนาดไม่มีฟังก์ชันติดต่อ Database และ Form Validation ที่ Web Application ส่วนใหญ่

ต้องใช้ ติดตั้งง่าย มีความยืดหยุ่นสูง โดย Flask รองรับการเพิ่มความสามารถอื่นๆ ผ่านทาง Extension ที่มีให้เลือกจำนวนมากมาย พัฒนาโดยชุมชนขนาดใหญ่ที่มีความ Active

บทที่ 3

แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนการฝึกงาน

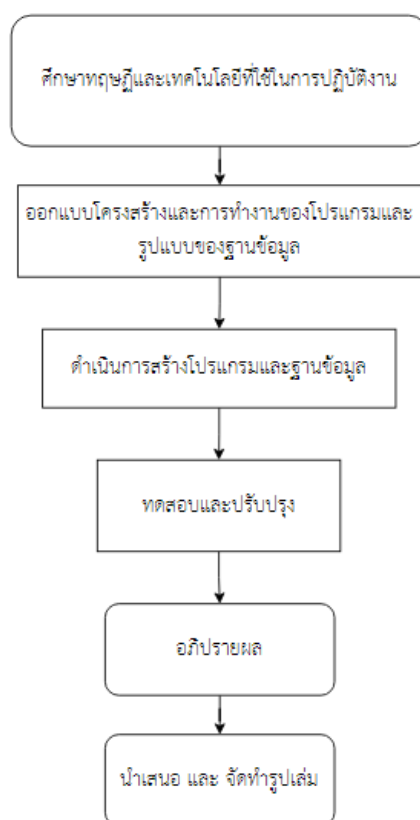
ในการฝึกงานได้มีการวางแผนกำหนดหัวข้อในการดำเนินงานจัดเรียงลำดับความสำคัญ กำหนดเวลาให้เหมาะสมกับหัวข้อและระยะเวลาการฝึกงาน เพื่อให้งานที่ได้รับมอบหมายสำเร็จลุล่วงตามวัตถุประสงค์ ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 แผนการฝึกงาน 1 กรกฎาคม 2564 – 1 สิงหาคม 2564

ลำดับที่	หัวข้อการดำเนินงาน	เดือนที่ 7				เดือนที่ 8			
		1	2	3	4	5	6	7	8
1	รับมอบหมายหัวข้อการศึกษาในการฝึกงาน								
2	ศึกษาทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน								
3	ศึกษาแนวทางพัฒนาเพื่อนำไปประยุกต์ใช้								
4	สร้างระบบด้วยภาษา Python โดยใช้ Flask Framework ร่วมกับ Cloud database								
5	ทดสอบการนำไปใช้งานและปรับปรุง								
6	สรุปผลการทดสอบ								
7	นำเสนอและจัดทำรูปเล่ม								

3.2 ขั้นตอนการกำเนินโครงการ

การศึกษาเรื่องเว็บแอปพลิเคชันสำหรับจัดการอุปกรณ์อำนวยความสะดวกในห้องปฏิบัติการ การ ลำดับขั้นตอนในการดำเนินงานสามารถแสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการดำเนินงาน

3.3 ขั้นตอนการพัฒนาโปรแกรมส่วนการประมวลผลฝั่งเซิร์ฟเวอร์ด้วยภาษา Python และ Flask Framework

การจัดการและการดำเนินการของระบบนั้นส่วนใหญ่จะเกิดขึ้นในฝั่งของเซิร์ฟเวอร์ และ จากการศึกษาจึงทำให้ผู้ทำงานวิจัยนั้นได้เลือกพัฒนาระบบขึ้นมาจากภาษา Python และใช้ Flask Framework โดยมี Extensions ต่างที่ใช้ในการพัฒนาดังแสดงในรูปที่ 3.2

```

from flask import Flask , render_template, redirect, url_for, request
from flask_login.utils import login_required
from flask_sqlalchemy import SQLAlchemy
from flask_marshmallow import Marshmallow
from flask_bootstrap import Bootstrap
from flask_wtf import FlaskForm
from wtforms import StringField, BooleanField, TextAreaField
from wtforms.fields.core import SelectField
from wtforms.validators import InputRequired, Length, Email
from flask_login import LoginManager, UserMixin, login_manager, login_user, logout_user, current_user
import serverDB

```

รูปที่ 3.2 รายการ Extension ที่จำเป็นในการเขียนสร้างระบบจัดการอุปกรณ์

ในส่วนของ cloud database นั้นได้ใช้บริการ database ของ FreeSQLdatabase โดยมี MySQL เป็นระบบฐานข้อมูลและมีการตั้งค่าการเชื่อมต่อกับ database ดังรูปที่ 3.3

```

conn = f'mysql+pymysql://{serverDB.db_user}:{serverDB.db_password}@{serverDB.db_host}:{serverDB.db_port}/{serverDB.db_name}'
# conn = 'sqlite:///database.db'

app = Flask(__name__)
app.config['SECRET_KEY'] = 'DREAM_LabsProjectDB'
app.config['SQLALCHEMY_DATABASE_URI'] = conn
db = SQLAlchemy(app)
ma = Marshmallow(app)
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'

Bootstrap(app)

```

รูปที่ 3.3 การตั้งค่าการเชื่อมต่อกับ Cloud database และการเริ่มต้นการทำงานของ Extension ต่างๆ

โครงสร้างของ database จะแบ่งข้อมูลเป็นสองชุดโดยชุดแรกจะเก็บเป็นข้อมูลของอุปกรณ์ในห้องปฏิบัติการและอีกชุดจะเป็นชุดข้อมูลในส่วนของสมาชิกในห้องปฏิบัติการโดยโครงสร้างเป็นไปดังรูปที่ 3.4


```
# database model structure

class equipment_database(db.Model):
    __table_args__ = ( 'mysql_engine': 'InnoDB', 'mysql_charset': 'utf8', 'mysql_collate': 'utf8_general_ci' )
    id = db.Column(db.Integer, primary_key=True) #Ex. '1','2'
    device = db.Column(db.String(50), unique=True, nullable=False) #Ex. 'DEREE DE-208E', 'POWERS0001'
    type = db.Column(db.String(50), nullable=False) #Ex. 'Multimeter', 'PowerSupply'
    serial = db.Column(db.String(50), unique=True, nullable=False) #Ex. 'N00016141', 'PS000001'
    status = db.Column(db.String(50), db.ForeignKey('user_database.name')) #Ex. 'Available' or 'user.name' *In use*
    description = db.Column(db.String(255))
    device_user = db.relationship('user_database', backref='usingdevice')

    def __repr__(self):
        return f'id: {self.id} | device: {self.device} | serial: {self.serial} | status: {self.status} | description: {self.description} |'

class user_database(UserMixin, db.Model):
    __table_args__ = ( 'mysql_engine': 'InnoDB', 'mysql_charset': 'utf8', 'mysql_collate': 'utf8_general_ci' )
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(80), unique=True, nullable=False) #Ex. 'Netithorn Preechapattananont', 'Available', 'Unavailable'
    email = db.Column(db.String(50), unique=True, nullable=False) #Ex. 'pr.netithorn_st@tni.ac.th', 'a.dream_lab@tni.ac.th', 'u.dream_lab@tni.ac.th'
    tni_id = db.Column(db.String(10), unique=True, nullable=False) #Ex. '60115041-0', '111111', '000000'
    type = db.Column(db.String(50), nullable=False) #Ex. 'Student', 'Other', 'Other'
    description = db.Column(db.String(255)) #Ex. '', 'Available status', 'Unavailable status'

@login_manager.user_loader
def load_user(user_id):
    return user_database.query.get(int(user_id))
```

รูปที่ 3.4 โครงสร้างและข้อกำหนดของฐานข้อมูลอุปกรณ์ และ สมาชิก

ตารางของชุดข้อมูลอุปกรณ์ในห้องปฏิบัติการนั้นจะเก็บข้อมูลในส่วนของ ชื่ออุปกรณ์ ชนิดของอุปกรณ์ Serial Number สถานะของอุปกรณ์ และ คำอธิบายเพิ่มเติมดังรูปที่ 3.5

id	device	type	serial	status	description
1	multimeter_1	multimeter	N00016141	TestTeacher_1	NULL
4	power supply_1	power supply	pwr12412	Unavailable	NULL
6	testdevice_1	Other	testdevice_serial_1	Available	NULL
7	testdevice_2	Multimeter	testdevice_serial_2	TestTeacher_1	NULL
8	testdevice_3	Multimeter	testdevice_serial_3	Netithorn Preechapattananont	NULL
9	testdevice_4	Oscilloscope	testdevice_serial_4	Available	NULL
10	testdevice_5	Oscilloscope	testdevice_serial_5	Available	NULL
11	testdevice_6	Oscilloscope	testdevice_serial_6	TestTeacher_1	NULL
12	testdevice_7	Oscilloscope	testdevice_serial_7	TestTeacher_1	NULL
13	testdevice_8	MCU	testdevice_serial_8	Available	NULL
14	testdevice_9	MCU	testdevice_serial_9	Available	NULL
25	addtest_1	Multimeter	TESTADD	Available	
26	addtest_2	Multimeter	TESTADD2	Available	some description

รูปที่ 3.5 ตัวอย่างข้อมูลอุปกรณ์ที่ถูกจัดเก็บอยู่บน Cloud database

ในส่วนของตารางของชุดข้อมูลสมาชิกในห้องปฏิบัติการนั้นจะเก็บข้อมูลในส่วนของ ชื่อสมาชิก e-mail สถาบันของสมาชิก รหัสประจำตัวสถาบันของสมาชิก สถานะของสมาชิก และ คำอธิบายเพิ่มเติมดังรูปที่ 3.6

id	name	email	ttni_id	type	description
1	Netithorn Preechapattananont	pr.netithorn_st@ttni.ac.th	123456	Student	admin
2	TestTeacher_1	testteacher@gmail.com	654321	Teacher	NULL
3	Available	a.dream_lab@ttni.ac.th	111111	Other	Available status
4	Unavailable	u.dream_lab@ttni.ac.th	000000	Other	Unavailable status
6	Chanapong Aiamraksa	ai.chanapong_st@ttni.ac.th	60115050-1	Student	NULL

รูปที่ 3.6 ตัวอย่างข้อมูลสมาชิกที่ถูกจัดเก็บอยู่บน Cloud database

รูปแบบของการเก็บข้อมูลจะเก็บข้อมูลดังที่ได้กล่าวมาข้างต้นโดยเป็นไปตามรูปแบบและเงื่อนไข ดังรูปที่ 3.7 เพื่อนำไปบันทึกลง cloud database และ ประเมินผลต่อไป

```
# form class

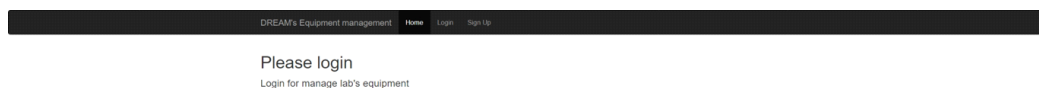
class loginForm(FlaskForm):
    email = StringField('E-mail', validators=[InputRequired(), Email(message='Please Enter your TTI E-mail'), Length(min=5, max=40)])
    ttni_id = StringField('TTI ID', validators=[InputRequired(), Length(min=6, max=80)])
    remember = BooleanField('Remember Me')

class registerForm(FlaskForm):
    email = StringField('E-mail', validators=[InputRequired(message='Please enter your E-mail'), Email(message='Please Enter your TTI E-mail'), Length(min=5, max=40)])
    ttni_id = StringField('TTI ID', validators=[InputRequired(message='Please enter your TTI ID'), Length(min=6, max=80)])
    name = StringField('Name', validators=[InputRequired(message='Please enter your name')])
    type = SelectField('Member position', choices=[('Student', 'Student'), ('Teacher', 'Teacher'), ('Other', 'Other')], validators=[InputRequired(message='Please choose your position')])
    description = TextAreaField('Description')

class EquipmentForm(FlaskForm):
    device = StringField('Device', validators=[InputRequired()])
    type = SelectField('Device Type', choices=[('Multimeter', 'Multimeter'), ('Power Supply', 'Power Supply'), ('Oscilloscope', 'Oscilloscope'), ('Signal generator', 'Signal generator'), ('MCU', 'MCU'), ('Sensor', 'Sensor')])
    serial = StringField('Serial', validators=[InputRequired()])
    status = SelectField('Status', choices=[('Available', 'Available'), ('Unavailable', 'Unavailable')], validators=[InputRequired(message='Please choose status')])
    description = TextAreaField('Description')
```

รูปที่ 3.7 รูปแบบ Form ที่มีเงื่อนไขสำหรับเก็บข้อมูลของการ login sign-up และ อุปกรณ์

ทางด้านของการประมวลผลของฝั่งเซิร์ฟเวอร์นั้นจะแบ่งออกเป็นการทำงานในส่วนของหน้าเว็บย่อยต่างๆเช่นการทำงานในส่วนของ index login และ sign-up เป็นต้น ในส่วนของหน้า index นั้นจะเป็นหน้าเริ่มต้นที่จะให้ผู้ใช้ทำการ login หรือ sign-up เพื่อใช้งานต่อไปโดยหน้าเว็บจะเป็นดังรูปที่ 3.8



รูปที่ 3.8 หน้าต่างของ index ของระบบจัดการอุปกรณ์

ในการทำงานในส่วนของการ login นั้นจะทำการเก็บข้อมูลโดยใช้รูปแบบการเก็บข้อมูลในส่วน ของ login ที่ได้กล่าวมาในข้างต้นหลังจากนั้นเมื่อผู้ใช้ได้กด submit ข้อมูลมาแล้วระบบจะทำการเช็ค ว่า e-mail ที่ได้กรอกหน้า login นั้นได้ตรงกับข้อมูล e-mail ที่ได้ถูกเก็บไว้ในฐานข้อมูลหรือไม่หลังจาก นั้นจะเช็คในส่วนของรหัสประจำตัวที่ถูกผูกไว้กับตัว e-mail นั้นว่าตรงหรือไม่ถ้าถูกต้องระบบจะทำ การ redirect ไปหน้าของ dashboard ต่อไปถ้าระบบตรวจสอบแล้วพบว่าไม่ตรงระบบจะแจ้งว่า login ไม่ สำเร็จเนื่องจาก e-mail หรือ รหัสประจำตัวผิด โดยการทำงานทั้งหมดจะเป็นไปตามโค้ดดังรูปที่ 3.9 ใน ส่วนของหน้าการแสดงผลในฝั่งของผู้ใช้งานจะเป็นไปตามรูปที่ 3.10

```
#Site Route

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    form = loginForm()

    if form.validate_on_submit():
        email = user_database.query.filter_by(email=form.email.data).first()
        if email:
            if email.tni_id == form.tni_id.data:
                login_user(email, remember=form.remember.data)
                return redirect(url_for('dashboard'))

            return '<h1> Invalid email or tni_id </h1>'

    return render_template('login.html', form=form)
```

รูปที่ 3.9 โค้ดการทำงานในฝั่งของเซิร์ฟเวอร์ในหน้าของ index และ login



รูปที่ 3.10 หน้าต่าง login

ส่วนการแสดงผลและการประมวลผลฝั่งผู้ใช้นั้นจะมีความทำงานโดยเก็บข้อมูลในส่วนของ e-mail รหัสประจำตัวสถาบัน และการจดจำการ login ดังรูปที่ 3.11 ก่อนที่จะส่งไปประมวลผลที่ฝั่งเซิร์ฟเวอร์ดังที่กล่าวไปแล้วข้างต้น

```
{% extends "bootstrap/base.html" %}
{% import "bootstrap/wtf.html" as wtf %}

{% block title %}
Login
{% endblock %}

{% block styles %}
{{super()}}
<link rel="stylesheet" href="{{url_for('.static', filename='signin.css')}}">
{% endblock %}

{% block content %}
<div class="container">
  <form class="form-signin" method="POST" action="/login">
    <h2 class="form-signin-heading">Please sign in</h2>
    {{ form.hidden_tag() }}
    {{ wtf.form_field(form.email) }}
    {{ wtf.form_field(form.tni_id) }}
    {{ wtf.form_field(form.remember) }}
    <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
  </form>
</div> <!-- /container -->
{% endblock %}
```

รูปที่ 3.11 โค้ด html ในส่วนของหน้าต่าง login

ในกรณีที่ผู้ใช้นั้นยังไม่มีข้อมูลสมาชิกในฐานข้อมูลนั้นจำเป็นจะต้องเพิ่มข้อมูลลงในฐานข้อมูลเสียก่อนโดยหลักการทำงานจะคล้ายๆกับการ login โดยจะเก็บข้อมูลตามรูปแบบและเงื่อนไขตามที่ได้ออกแบบฐานข้อมูลไว้ก่อนหน้าโดยมีการเก็บข้อมูลตาม form ดังรูปที่ 3.14 หลังจากนั้นเมื่อผู้ใช้

กด submit ระบบจะทำการบันทึกข้อมูลที่ใช้ได้ระบุลงในฐานข้อมูลและจะทำการ redirect ไปยังหน้า login

```
@app.route('/signup', methods=['GET', 'POST'])
def signup():
    form = registerForm()

    if form.validate_on_submit():
        # hashed_password = generate_password_hash(form.password.data, method='sha256')
        # new_user = user_database(name=form.name.data, email=form.email.data, password=hashed_password, type=form.type.data)
        new_user = user_database(name=form.name.data, email=form.email.data, tni_id=form.tni_id.data, type=form.type.data)
        db.session.add(new_user)
        db.session.commit()
        # return '<h1> New user has been create </h1>'
        return redirect(url_for('login'))

    return render_template('signup.html', form=form)
```

รูปที่ 3.12 โค้ดการทำงานในฝั่งของเซิร์ฟเวอร์ในหน้าของ signup

```
{% extends "bootstrap/base.html" %}
{% import "bootstrap/wtf.html" as wtf %}

{% block title %}
Sign Up
{% endblock %}

{% block styles %}
{{super()}}
<link rel="stylesheet" href="{{url_for('.static', filename='signin.css')}}">
{% endblock %}

{% block content %}
<div class="container">

    <form class="form-signin", method="POST" action="/signup">
        <h2 class="form-signin-heading">Sign Up</h2>
        {{ form.hidden_tag() }}
        {{ wtf.form_field(form.name) }}
        {{ wtf.form_field(form.tni_id) }}
        {{ wtf.form_field(form.email) }}
        {{ wtf.form_field(form.type) }}
        {{ wtf.form_field(form.description) }}
        <button class="btn btn-lg btn-primary btn-block" type="submit">Sign Up</button>
    </form>

</div> <!-- /container -->
{% endblock %}
```

รูปที่ 3.13 โค้ด html ในส่วนของหน้าต่าง signup

รูปที่ 3.14 หน้าต่าง signup

หลังจากทำการ login ได้สำเร็จระบบจะทำการ redirect ไปยังหน้า dashboard ดังรูปที่ 3.15 โดยระบบจะทำการเช็คสิทธิ์ว่าผู้ที่ทำการ login นั้นเป็น admin หรือไม่เพื่อที่จะมอบสิทธิ์ในการจัดการแก้ไข และ เพิ่มอุปกรณ์ ให้กับผู้ใช้ต่อไป โดยในหน้า dashboard นั้นจะเป็นหน้าทำการหลักของระบบที่จะสามารถทำให้ผู้ใช้ระบุสถานะของอุปกรณ์ได้ว่าจะยืมหรือคืนอุปกรณ์โดยจะทำการฐานข้อมูลของอุปกรณ์มาแสดงเมื่อผู้ใช้เลือกประเภทของอุปกรณ์ที่ต้องการให้ระบบแสดงโดยแยกประเภท เช่น My Workbench ที่จะแสดงเฉพาะอุปกรณ์ที่ผู้ใช้กำลังใช้งานอยู่ All devices ที่จะแสดงอุปกรณ์ทั้งหมดที่อยู่ในฐานข้อมูล หรือ Multimeter ที่จะทำการแสดงเฉพาะอุปกรณ์ประเภท Multimeter เป็นต้น

ID	Device	Type	Serial	Status	Description	Pick/Return
1	multimeter_1	multimeter	N60016141	TestTeacher_1	None	Pick
4	power supply_1	power supply	port12412	Unavailable	None	Unavailable
6	testdevice_1	Other	testdevice_serial_1	Available	None	Pick
7	testdevice_2	Multimeter	testdevice_serial_2	TestTeacher_1	None	Pick
8	testdevice_3	Multimeter	testdevice_serial_3	Netthom Preechapattananont	None	Pick
9	testdevice_4	Oscilloscope	testdevice_serial_4	Netthom Preechapattananont	None	Pick
10	testdevice_5	Oscilloscope	testdevice_serial_5	Netthom Preechapattananont	None	Pick
11	testdevice_6	Oscilloscope	testdevice_serial_6	TestTeacher_1	None	Pick
12	testdevice_7	Oscilloscope	testdevice_serial_7	TestTeacher_1	None	Pick
13	testdevice_8	MCU	testdevice_serial_8	Available	None	Pick
14	testdevice_9	MCU	testdevice_serial_9	Available	None	Pick
25	adddtest_1	Multimeter	TESTADD1	Available	None	Pick
26	adddtest_2	Multimeter	TESTADD2	Available	some description	Pick

รูปที่ 3.15 หน้าต่าง dashboard ของผู้ใช้งาน (สมาชิกธรรมดา)

ในส่วนของการทำงานในฝั่งของเซิร์ฟเวอร์การดึงข้อมูลที่จะมาแสดงนั้นจะต้องทำการรับข้อมูลมาจากฝั่งของผู้ใช้เพื่อมาทำการเลือกดึงเฉพาะข้อมูลที่ต้องการจากฐานข้อมูลจากนั้นระบบจะรอข้อมูลการ

ยืมหรือการคืนอุปกรณ์จากผู้ใช้อีกต่อไปเพื่อที่จะนำข้อมูลนั้นไปแก้ไขให้กับสถานะของอุปกรณ์นั้นในฐานข้อมูลต่อไป

```
@app.route('/dashboard', methods=['GET', 'POST'])
@login_required
def dashboard():

    user_name = current_user.name
    admin_check = user_database.query.filter_by(name=user_name).first().description
    data = equipment_database.query.all()
    filter = request.args.get('radio')

    if filter == 'All Devices' :
        data = equipment_database.query.all()
    elif filter == 'My Workbench':
        data = equipment_database.query.filter_by(status=user_name).all()
    else:
        data = equipment_database.query.filter_by(type=filter).all()

    if request.method == 'POST':

        action = request.form.get('subbt')
        deviceid = request.form.get('deviceid')
        data = equipment_database.query.filter_by(status=user_name).all()
        filter = request.form.get('filter')

        if filter == 'All' :
            data = equipment_database.query.all()
        elif filter == 'My':
            data = equipment_database.query.filter_by(status=user_name).all()
        elif filter == 'Other':
            data = equipment_database.query.filter_by(type='Other').all()
        elif filter == 'Signal':
            data = equipment_database.query.filter_by(type='Signal generator').all()
        else:
            data = equipment_database.query.filter_by(type=filter).all()

        if action == 'Pick':
            device_update = equipment_database.query.filter_by(id=deviceid).first()
            device_update.status = current_user.name
            db.session.commit()
        elif action == 'Return':
            device_update = equipment_database.query.filter_by(id=deviceid).first()
            device_update.status = 'Available'
            db.session.commit()

    return render_template('dashboard.html', name = current_user.name, data = data, filter = filter, permission = admin_check )
```

รูปที่ 3.16 โค้ดการทำงานในฝั่งของเซิร์ฟเวอร์ในหน้าของ dashboard ที่มีการดึงข้อมูลและแก้ไขข้อมูลสถานะไปยัง Cloud database

ด้านการทำงานในฝั่งของผู้ใช้งานระบบจะทำการเก็บข้อมูล keyword ดังรูปที่ 3.17 แล้วทำการส่งไปยังฝั่งของเซิร์ฟเวอร์เพื่อให้ทางฝั่งเซิร์ฟเวอร์นำไปใช้คัดกรองข้อมูลก่อนที่จะดึงข้อมูลมาจากฐานข้อมูลแล้วส่งไปแสดงผลมาฝั่งผู้ใช้อีกดังรูปที่ 3.18 เพื่อที่จะทำการยืม และ คืนต่อไป

ในกรณีที่ผู้ใช้มีสิทธิ์เป็น admin ในหน้า dashboard จะปรากฏสถานะ admin และ หัวข้อ Manage devices ดังรูป 3.19

ID	Device	Type	Serial	Status	Description	Pick/Return
1	multimeter_1	multimeter	N00016141	TestTeacher_1	None	in use
4	power supply_1	power supply	psw12412	Unavailable	None	Unavailable
6	testdevice_1	Other	testdevice_serial_1	Available	None	in use
7	testdevice_2	Multimeter	testdevice_serial_2	TestTeacher_1	None	in use
8	testdevice_3	Multimeter	testdevice_serial_3	Netithorn Preechapattananont	None	in use
9	testdevice_4	Oscilloscope	testdevice_serial_4	Netithorn Preechapattananont	None	in use
10	testdevice_5	Oscilloscope	testdevice_serial_5	Netithorn Preechapattananont	None	in use
11	testdevice_6	Oscilloscope	testdevice_serial_6	TestTeacher_1	None	in use
12	testdevice_7	Oscilloscope	testdevice_serial_7	TestTeacher_1	None	in use
13	testdevice_8	MCU	testdevice_serial_8	Available	None	in use
14	testdevice_9	MCU	testdevice_serial_9	Available	None	in use
25	adddst_1	Multimeter	TESTADD1	Available	None	in use
26	adddst_2	Multimeter	TESTADD2	Available	some description	in use

รูปที่ 3.19 หน้าต่าง dashboard ของผู้ใช้งาน (admin)

เมื่อเข้ามายังหัวข้อ Manage devices แล้วจะปรากฏหน้าต่างดังรูป 3.20 โดยพื้นฐานการแสดงผลจะเป็นเช่นเดียวกับหน้า dashboard แต่การทำงานจะแตกต่างกันโดยในหน้านี้จะสามารถแก้ไข ลบ หรือ เพิ่ม อุปกรณ์ได้

ID	Device	Type	Serial	Status	Description	Remove/Edit
1	multimeter_1	multimeter	N00016141	TestTeacher_1	None	Add Remove
4	power supply_1	power supply	psw12412	Unavailable	None	Add Remove
6	testdevice_1	Other	testdevice_serial_1	Available	None	Add Remove
7	testdevice_2	Multimeter	testdevice_serial_2	TestTeacher_1	None	Add Remove
8	testdevice_3	Multimeter	testdevice_serial_3	Netithorn Preechapattananont	None	Add Remove
9	testdevice_4	Oscilloscope	testdevice_serial_4	Netithorn Preechapattananont	None	Add Remove
10	testdevice_5	Oscilloscope	testdevice_serial_5	Netithorn Preechapattananont	None	Add Remove
11	testdevice_6	Oscilloscope	testdevice_serial_6	TestTeacher_1	None	Add Remove
12	testdevice_7	Oscilloscope	testdevice_serial_7	TestTeacher_1	None	Add Remove
13	testdevice_8	MCU	testdevice_serial_8	Available	None	Add Remove
14	testdevice_9	MCU	testdevice_serial_9	Available	None	Add Remove
25	adddst_1	Multimeter	TESTADD1	Available	None	Add Remove
26	adddst_2	Multimeter	TESTADD2	Available	some description	Add Remove

รูปที่ 3.20 หน้าต่าง Manage devices สำหรับการจัดการอุปกรณ์

```

@app.route('/dashboard/ad_manage', methods=['GET', 'POST'])
@login_required
def admin_manage():
    form = EquipmentForm()

    user_name = current_user.name
    admin_check = user_database.query.filter_by(name=user_name).first().description
    data = equipment_database.query.all()
    filter = request.args.get('radio')

    if filter == 'All Devices' :
        data = equipment_database.query.all()
    else:
        data = equipment_database.query.filter_by(type=filter).all()

    if request.method == 'POST':
        action = request.form.get('subbt')
        deviceid = request.form.get('deviceid')
        filter = request.form.get('filter')

        if filter == 'All' :
            data = equipment_database.query.all()
        elif filter == 'Other':
            data = equipment_database.query.filter_by(type='Other').all()
        elif filter == 'Signal':
            data = equipment_database.query.filter_by(type='Signal generator').all()
        else:
            data = equipment_database.query.filter_by(type=filter).all()

        if action == 'Edit':
            device_edit = equipment_database.query.filter_by(id=deviceid).first()
            device_edit.device = request.form.get('device')
            device_edit.type = request.form.get('type')
            device_edit.serial = request.form.get('serial')
            device_edit.status = request.form.get('status')
            device_edit.description = request.form.get('description')
            db.session.commit()
        pass
        elif action == 'Add':
            new_device = equipment_database(device=form.device.data , type=form.type.data, serial=form.serial.data, status=form.status.data, description = form.description.data)
            db.session.add(new_device)
            db.session.commit()
        pass
        elif action == 'Remove':
            device_delete = equipment_database.query.filter_by(id=deviceid).first()
            db.session.delete(device_delete)
            db.session.commit()
        return redirect(url_for('admin_manage'))

```

รูปที่ 3.21 โค้ดการทำงานของฝั่งเซิร์ฟเวอร์ในหน้าของ Manage devices ที่มีการดึงข้อมูล แก้ไขข้อมูล และ เพิ่มข้อมูลไปยัง Cloud database

ในส่วนของการเพิ่มและแก้ไขข้อมูลของอุปกรณ์นั้นจะคล้ายคลึงกันเพียงแต่การเพิ่มอุปกรณ์นั้นจะทำให้ผู้ใช้กรอกข้อมูลของอุปกรณ์ทั้งหมดดังรูปที่ 3.22 หลังจากนั้นจึงส่งไปยังฝั่งเซิร์ฟเวอร์เพื่อทำการเพิ่มข้อมูลเข้าไปยังฐานข้อมูลโดยมีการทำงานดังรูปที่ 3.23 ต่อไปส่วนในกรณีที่จะทำการแก้ไขข้อมูลของอุปกรณ์นั้นจะทำการดึงข้อมูลของอุปกรณ์ชิ้นนั้นๆมาก่อนหลังจากนั้นค่อยให้ผู้ใช้แก้ไขดังรูปที่ 3.24 แล้วจึงส่งไปยังฝั่งเซิร์ฟเวอร์เพื่อแก้ไขข้อมูลในฐานข้อมูลต่อไปดังรูปที่ 3.25 และ 3.26

รูปที่ 3.22 หน้าต่าง pop-up form ในการเพิ่มอุปกรณ์

```

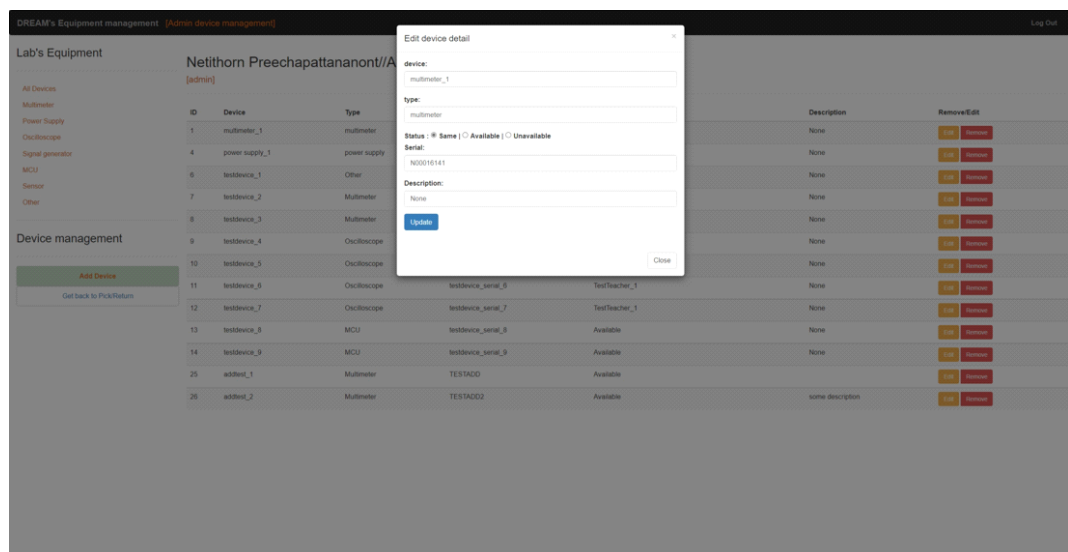
<hr>
{% if permission == 'admin' %}
<li style="font-weight: 500; font-size: x-large;">Device management</li>

<hr>
<li><a data-toggle="modal" data-target="#modaledit" class="btn btn-default" style="font-weight: 600; color: chocolate; background-color: #e6f2ff;" href="#">Add Device</a></li>
<div id="modaledit" class="modal fade" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal">&times;</button>
<h4 class="modal-title">Device detail</h4>
</div>
<div class="modal-body">
<form action="{url_for('admin_manage')}}" method="POST">
<div class="form-group">
{{ form.hidden_tag() }}
{{ wtf.form_field(form.device) }}
</div>
<div class="form-group">
{{ wtf.form_field(form.type) }}
</div>
<div>
{{ wtf.form_field(form.status) }}
</div>
<div class="form-group">
{{ wtf.form_field(form.serial) }}
</div>
<div class="form-group">
{{ wtf.form_field(form.description) }}
</div>
<div class="form-group">
<input type="hidden" name="filter" value="{{filter}}">
<button href="{url_for('admin_manage')}}" class="btn btn-primary" type="submit" name="subbt" value='Add'>Add</button>
<script></script>
</div>
</form>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
</div>
</div>
</div>
</div>

<li><a class="btn btn-default" style="color: #337ab7;" href="/dashboard">Get back to Pick/Return </a></li>
<!-- <li><a href="#" style="color: chocolate;">Manage member</a></li> -->
{% endif %}

```

รูปที่ 3.23 โค้ด html ในหน้าต่าง Manage devices ในส่วนของ pop-up form ที่เก็บข้อมูลของอุปกรณ์ก่อนส่งข้อมูลไปต่อยังการประมวลผลฝั่งเซิร์ฟเวอร์เพื่อเพิ่มข้อมูล



รูปที่ 3.24 หน้าต่าง pop-up form ในการแก้ไขข้อมูลของอุปกรณ์

```
{% block content %}
<div>
<h2 class="sub-header"> {{ name }}/{{filter}} </h2>{% if permission == 'admin' %}<h4 style="color: #chocolate;"> [{{ permission }}]</h4>{% endif %}
<hr>
<!-- <h3 class="sub-header">{{ filter }}</h3> -->
<div class="table-responsive">
<table class="table table-striped">
<thead>
<tr>
<th>ID</th>
<th>Device</th>
<th>Type</th>
<th>Serial</th>
<th>Status</th>
<th>Description</th>
<th>Remove/Edit</th>
</tr>
</thead>
<tbody>
{% for row in data %}
<tr>
<td>{{ row.id }}</td>
<td>{{ row.device }}</td>
<td>{{ row.type }}</td>
<td>{{ row.serial }}</td>
<td>{{ row.status }}</td>
<td>{{ row.description }}</td>
<td>
<button class="btn btn-warning btn-sm" data-toggle='modal' data-target='#modaledit{{row.id}}'>Edit</button>
<div id="modaledit{{row.id}}" class="modal fade" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal">&times;</button>
<h4 class="modal-title">Edit device detail</h4>
</div>
<div class="modal-body">
<form action="{% url_for('admin_manage') %}" method="POST">
<div class="form-group">
<label>device:</label>
<input type="hidden" name="id" value="{{row.id}}">
<input type="text" class="form-control" name="device" value="{{row.device}}">
</div>
<div class="form-group">
<label>type:</label>
<input type="text" class="form-control" name="type" value="{{row.type}}">
</div>

```

รูปที่ 3.25 โค้ด html ในหน้าต่าง Manage devices ในส่วนของ pop-up form ที่เก็บข้อมูลของอุปกรณ์ก่อนส่งข้อมูลไปต่อยังการประมวลผลฝั่งเซิร์ฟเวอร์เพื่อแก้ไขข้อมูล

```

<div>
  <label>Status : </label>
  <input type="radio" id="not_change" name="status" value="{{row.status}}" checked>
  <label>Same | </label>
  <span></span>
  <input type="radio" id="available" name="status" value="Available">
  <label for="available">Available | </label>
  <span></span>
  <input type="radio" id="unavailable" name="status" value="Unavailable">
  <label for="unavailable">Unavailable</label>
</div>
<div class="form-group">
  <label>Serial:</label>
  <input type="text" class="form-control" name="serial" value="{{row.serial}}">
</div>
<div class="form-group">
  <label>Description:</label>
  <input type="text" class="form-control" name="decription" value="{{row.description}}">
</div>
<div class="form-group">
  <input type="hidden" name="deviceid" value={{row.id}}>
  <input type="hidden" name="filter" value={{filter}}>
  <button class="btn btn-primary" type="submit" name='subbt' value='Edit'>Update</button>
</div>
</form>
</div>
<div class="modal-footer">
  <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
</div>
</div>
</div>
</div>
<form class="btn-group" role="group" action="{{url_for('admin_manage')}}" method="POST">
  <input type="hidden" name="deviceid" value={{row.id}}>
  <input type="hidden" name="filter" value={{filter}}>
  <button class="btn btn-danger btn-sm" type="submit" name='subbt' value='Remove'>Remove</button>
</form>
</form>
</td>
</tr>
{% endfor %}

```

รูปที่ 3.26 โค้ด html ในหน้าต่าง Manage devices ในส่วนของ pop-up form ที่เก็บข้อมูลของอุปกรณ์ก่อนส่งข้อมูลไปต่อยังการประมวลผลฝั่งเซิร์ฟเวอร์เพื่อแก้ไขข้อมูล (ต่อ)

บทที่ 4

สรุปผลการดำเนินงาน การวิเคราะห์และสรุปต่างๆ

4.1 ขั้นตอนและผลการดำเนินงาน

ขั้นตอนการดำเนินงาน

- 4.1.1.1 ออกแบบรูปแบบการทำงานของโปรแกรม
- 4.1.1.2 ออกแบบฐานข้อมูลที่ต้องใช้ร่วมกับโปรแกรม
- 4.1.1.3 ดำเนินการเขียนโปรแกรมในส่วนของการประมวลผลฝั่งเซิร์ฟเวอร์
- 4.1.1.4 ดำเนินการเขียนโปรแกรมในส่วนของการเชื่อมต่อกับ cloud database
- 4.1.1.5 ดำเนินการเขียนโปรแกรมในส่วนของการแสดงผล และการประมวลผลฝั่งผู้ใช้

ผลการดำเนินงาน

การเก็บข้อมูลผู้ใช้เข้าสู่ระบบนั้นจำเป็นที่จะต้องใส่ข้อมูล ชื่อ รหัสประจำตัวของสถาบัน e-mail และ สถานะของผู้ใช้ ในหน้า sign-up หลังจากผู้ใช้งานทำการ sign-up ดังรูปที่ 4.1 แล้วข้อมูลถูกเก็บไว้บน cloud database ดังรูปที่ 4.2

Sign Up

Name
signuptest name

TNI ID
112233

E-mail
signuptest@tni.ac.th

Member position
Student

Description
for sign up test

Sign Up

รูปที่ 4.1 การ sign-up

7	signuptest name	signuptest@tni.ac.th	112233	Student	NULL
---	-----------------	----------------------	--------	---------	------

รูปที่ 4.2 ข้อมูลผู้ใช้งานใน cloud database

ในการ login นั้นผู้ใช้จำเป็นต้องใส่ข้อมูล e-mail และ รหัสประจำตัวสถาบันตามที่ได้สมัครไว้ใน Sign-up ไว้ก่อนหน้า เมื่อผู้ใช้งานทำการ login ดังรูปที่ 4.3 จากนั้นจะเข้าสู่หน้า dashboard ดังรูปที่ 4.4 เพื่อใช้งานการยืม/คืนต่อไป

Please sign in

E-mail

signuptest@tni.ac.th

TNI ID

112233

☐ Remember Me

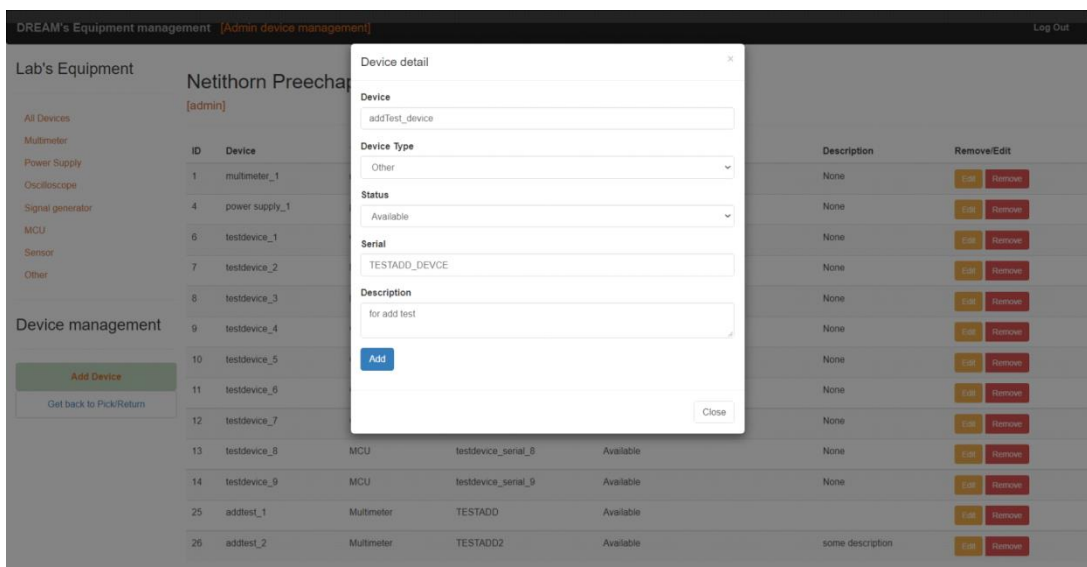
Sign in

รูปที่ 4.3 การ login เข้าสู่ระบบ

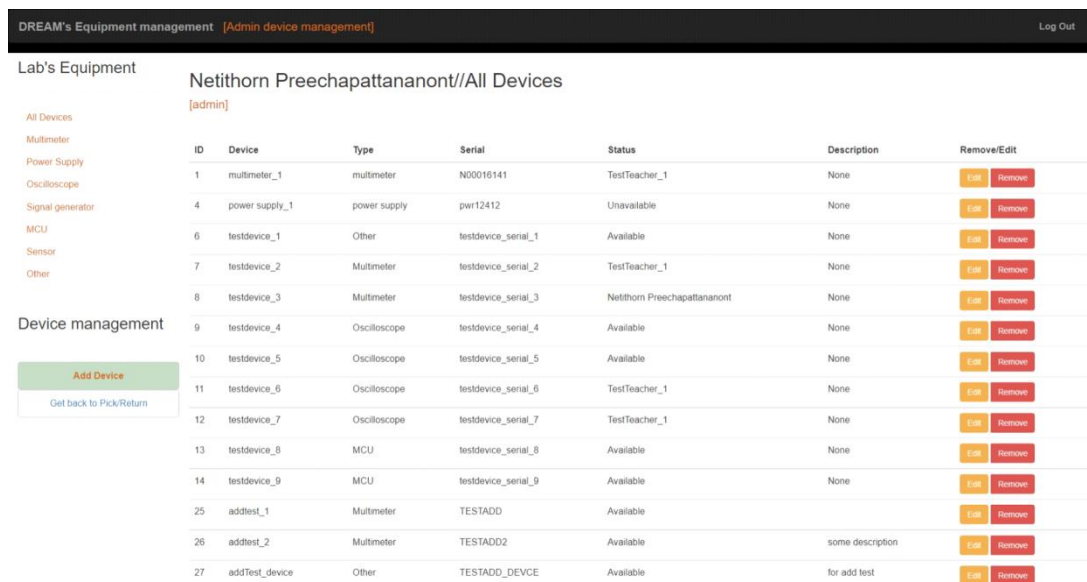
DREAM's Equipment management [Device Pick/Return] Log Out						
Lab's Equipment						
signuptest name						
<div>My Workbench</div> <div> All Devices Multimeter Power Supply Oscilloscope Signal generator MCU Sensor Other </div>						
ID	Device	Type	Serial	Status	Description	Pick/Return
1	multimeter_1	multimeter	N00016141	TestTeacher_1	None	In use
4	power supply_1	power supply	pwr12412	Unavailable	None	Unavailable
6	testdevice_1	Other	testdevice_serial_1	Available	None	Pick
7	testdevice_2	Multimeter	testdevice_serial_2	TestTeacher_1	None	In use
8	testdevice_3	Multimeter	testdevice_serial_3	Netthorn Preechapattananont	None	In use
9	testdevice_4	Oscilloscope	testdevice_serial_4	Available	None	Pick
10	testdevice_5	Oscilloscope	testdevice_serial_5	Available	None	Pick
11	testdevice_6	Oscilloscope	testdevice_serial_6	TestTeacher_1	None	In use
12	testdevice_7	Oscilloscope	testdevice_serial_7	TestTeacher_1	None	In use
13	testdevice_8	MCU	testdevice_serial_8	Available	None	Pick
14	testdevice_9	MCU	testdevice_serial_9	Available	None	Pick
25	addtest_1	Multimeter	TESTAD01	Available		Pick
26	addtest_2	Multimeter	TESTAD02	Available	some description	Pick

รูปที่ 4.4 หน้า dashboard สำหรับการใช้งาน ยืม/คืน อุปกรณ์

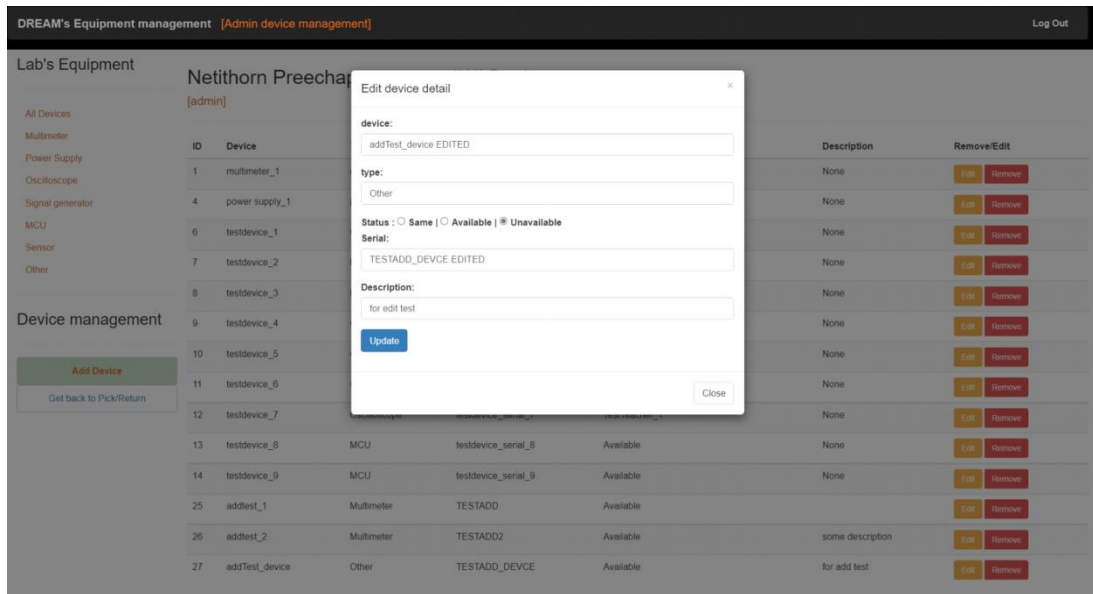
ในกรณีที่ผู้ใช้มีสถานะเป็น admin ทำให้ผู้ใช้สามารถเข้าสู่หน้าจัดการอุปกรณ์ได้ ผู้ใช้จึงสามารถ เพิ่มอุปกรณ์ในฐานข้อมูล แก้ไขข้อมูลของอุปกรณ์ และลบอุปกรณ์ออกจากฐานข้อมูลได้ ได้ดังรูปที่ 4.5 4.7 และ 4.9 ตามลำดับ



รูปที่ 4.5 การเพิ่มอุปกรณ์ในกรณีที่เป็น admin



รูปที่ 4.6 อุปกรณ์ถูกเพิ่ม



รูปที่ 4.7 การแก้ไขข้อมูลอุปกรณ์

DREAM's Equipment management [Admin device management] Log Out

Lab's Equipment

Netithorn Preechattanant//All [admin]

ID	Device	Type	Serial	Status	Description	Remove/Edit
1	multimeter_1	multimeter	N00016141	TestTeacher_1	None	Edit Remove
4	power supply_1	power supply	pwr12412	Unavailable	None	Edit Remove
6	testdevice_1	Other	testdevice_serial_1	Available	None	Edit Remove
7	testdevice_2	Multimeter	testdevice_serial_2	TestTeacher_1	None	Edit Remove
8	testdevice_3	Multimeter	testdevice_serial_3	Netithorn Preechattanant	None	Edit Remove
9	testdevice_4	Oscilloscope	testdevice_serial_4	Available	None	Edit Remove
10	testdevice_5	Oscilloscope	testdevice_serial_5	Available	None	Edit Remove
11	testdevice_6	Oscilloscope	testdevice_serial_6	TestTeacher_1	None	Edit Remove
12	testdevice_7	Oscilloscope	testdevice_serial_7	TestTeacher_1	None	Edit Remove
13	testdevice_8	MCU	testdevice_serial_8	Available	None	Edit Remove
14	testdevice_9	MCU	testdevice_serial_9	Available	None	Edit Remove
25	addtest_1	Multimeter	TESTADD	Available		Edit Remove
26	addtest_2	Multimeter	TESTADD2	Available	some description	Edit Remove
27	addTest_device EDITED	Other	TESTADD_DEVICE EDITED	Unavailable	None	Edit Remove

รูปที่ 4.8 ข้อมูลอุปกรณ์ถูกแก้ไข

DREAM's Equipment management [Admin device management] Log Out

Lab's Equipment

Netithorn Preechapattananont//All Devices
[admin]

All Devices
Multimeter
Power Supply
Oscilloscope
Signal generator
MCU
Sensor
Other

Device management

Add Device
Get back to Pick/Return

ID	Device	Type	Serial	Status	Description	Remove/Edit
1	multimeter_1	multimeter	N00016141	TestTeacher_1	None	<a>Edit <a>Remove
4	power supply_1	power supply	pw12412	Unavailable	None	<a>Edit <a>Remove
6	testdevice_1	Other	testdevice_serial_1	Available	None	<a>Edit <a>Remove
7	testdevice_2	Multimeter	testdevice_serial_2	TestTeacher_1	None	<a>Edit <a>Remove
8	testdevice_3	Multimeter	testdevice_serial_3	Netithorn Preechapattananont	None	<a>Edit <a>Remove
9	testdevice_4	Oscilloscope	testdevice_serial_4	Available	None	<a>Edit <a>Remove
10	testdevice_5	Oscilloscope	testdevice_serial_5	Available	None	<a>Edit <a>Remove
11	testdevice_6	Oscilloscope	testdevice_serial_6	TestTeacher_1	None	<a>Edit <a>Remove
12	testdevice_7	Oscilloscope	testdevice_serial_7	TestTeacher_1	None	<a>Edit <a>Remove
13	testdevice_8	MCU	testdevice_serial_8	Available	None	<a>Edit <a>Remove
14	testdevice_9	MCU	testdevice_serial_9	Available	None	<a>Edit <a>Remove
25	addtest_1	Multimeter	TESTADD	Available		<a>Edit <a>Remove
26	addtest_2	Multimeter	TESTADD2	Available	some description	<a>Edit <a>Remove

รูปที่ 4.9 การลบอุปกรณ์ออกจากฐานข้อมูล

โดยภายในหน้าต่างๆของระบบจะประกอบไปด้วยส่วนประกอบการทำงานต่างๆดังรูปที่ 4.10 และ 4.12 ตามลำดับ



รูปที่ 4.10 หัวข้อบน Nav-bar ในหน้า index

รูปที่ 4.11 ส่วนประกอบในหน้า Dashboard

The screenshot displays the 'DREAM's Equipment management' web application. The interface is divided into several sections:

- Top Bar:** Contains the title 'DREAM's Equipment management' and a 'Login Out' link.
- Left Sidebar:**
 - Lab's Equipment:** A list of equipment types: All Devices, Multimeter, Power Supply, Oscilloscope, Signal generator, MCU, Sensor, and Other.
 - Device management:** A section with an 'Add Device' button and a link to 'Get back to Pick/Return'.
- Main Content Area:**
 - Netithorn Preechapattananont//All Devices [admin]:** A header for the device list.
 - Table:** A table with columns: ID, Device, Type, Serial, Status, and Description. It lists 16 devices, including multimeters, power supplies, and oscilloscopes, with their respective serial numbers and statuses.
 - Right Panel:** A vertical list of device details, each with a 'Link' button and a 'Status' button.

Four red boxes highlight specific areas:

- Box 1: Points to the 'Lab's Equipment' sidebar.
- Box 2: Points to the 'Status' column in the device table.
- Box 3: Points to the 'Link' and 'Status' buttons in the right panel.
- Box 4: Points to the 'Login Out' link in the top bar.

รูปที่ 4.12 ส่วนประกอบในหน้า Manage devices

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

การจัดทำระบบการจัดการอุปกรณ์ของห้องปฏิบัติการ ระบบการแสดงสถานะ ยืม/คืน สามารถนำมาใช้ในการติดตามอุปกรณ์ที่ถูกนำไปใช้ได้ โดยระบบถูกพัฒนาขึ้นมาในรูปแบบของ Web application ที่ทำงานร่วมกับ Cloud database ซึ่งทำให้ผู้ใช้สามารถเข้าถึงได้ง่ายสามารถเข้าถึงจากที่ไหนก็ได้ที่สามารถต่อ Internet ได้แต่ในขณะเดียวกันหากผู้ใช้งานนั้นไม่มีความซื่อสัตย์และระเบียบวินัยต่อการระบุสถานะการ ยืม/คืน เมื่อได้ยืม/คืน อุปกรณ์ก็เป็นการยากในการติดตามอุปกรณ์เช่นเดิม

5.2 แนวทางการแก้ไขปัญหา

เนื่องจากปัจจุบันระบบนั้นจะทำการแก้ไขสถานะของอุปกรณ์ให้เป็นชื่อของผู้ที่เข้าใช้งานเท่านั้น ดังนั้นการแก้ปัญหาอาจทำได้โดยการใช้จัดตั้งผู้จัดการอุปกรณ์ผู้ใดผู้หนึ่งให้มีสิทธิ์ในการแก้ไขสถานะของอุปกรณ์แต่เพียงผู้เดียวส่วนผู้ใช้คนอื่นนั้นจำเป็นที่จะต้องต้องมายืม/คืนอุปกรณ์กับผู้จัดการอุปกรณ์เท่านั้น

5.3 ข้อเสนอแนะจากการดำเนินงาน

ควรที่จะมีผู้ที่ทำหน้าที่ดูแลฐานข้อมูลโดยตรงเพื่อที่จะสามารถจัดการฐานข้อมูล ติดตามขนาดของฐานข้อมูล และ คอยสังเกตถึงสิ่งข้อมูลที่ผิดปกติภายในฐานข้อมูล ในส่วนที่ admin ไม่สามารถทำได้

เอกสารอ้างอิง

- [1] Developer Team, **อยากเริ่มทำเว็บจากศูนย์ต้องรู้อะไรบ้าง ?**. [Online] ,
Available : <https://tinyurl.com/nabzhx9c> [Accessed: 3 กรกฎาคม 2564].
- [2] Developer Team.2020, **Bootstrap คืออะไร? ใช้งานยังไง?**. [Online] ,
Available : <https://tinyurl.com/3rtmr7hn> [Accessed: 3 กรกฎาคม 2564].
- [3] author,**Bootstrap5Tutorial**. [Online],
Available : <https://www.w3schools.com/bootstrap5/index.php>
[Accessed: 3 กรกฎาคม 2564].
- [4] วรินดา นวนกัน, **ฐานข้อมูล(Database)**. [Online],
Available : <https://tinyurl.com/3ehxja9s> [Accessed: 3 กรกฎาคม 2564].
- [5] author.2021, **SQL คืออะไร สำคัญอย่างไรต่อธุรกิจที่ขับเคลื่อนด้วย Data**. [Online]
, Available : <https://blog.pttexpresso.com/what-is-sql/>
[Accessed: 3 กรกฎาคม 2564].
- [6] วิทยากร 9Expert Training, **ภาษาโปรแกรม Python คืออะไร ?** . [Online] ,
Available : <https://tinyurl.com/ph98bpxc> [Accessed: 3 กรกฎาคม 2564].
- [7] saixiii.2017, **MySQL คืออะไร? และ ไรทำอะไร?** . [Online] ,
Available : <https://saixiii.com/what-is-mysql/> [Accessed: 3 กรกฎาคม 2564].
- [8] Keng Surapong.2020, **Flask คืออะไร พัฒนาเว็บ Web Application ด้วยภาษา Python สอนใช้งาน ตัวอย่าง Flask Framework ติดตั้ง บน Heroku by Example – heroku ep.2**. [Online], Available: <https://tinyurl.com/2svhxdwf> [Accessed: 3 กรกฎาคม 2564].

- [9] Laem.2011, .NET Framework คืออะไร มีที่มาและความสำคัญอย่างไร.
[Online],Available: <https://tinyurl.com/4r3f4sxn> [Accessed: 3 กรกฎาคม 2564].
- [10] author.2017, Framework คืออะไร.[Online]
Available : <https://tinyurl.com/4rkypf7b> [Accessed: 3 กรกฎาคม 2564].
- [11] Thai-Nichi Institute of Technology. 2005. แนะนำสถาบันเทคโนโลยีไทย-ญี่ปุ่น.
[Online],Available : <https://www.tni.ac.th/home/history/>.
[Accessed: 3 กรกฎาคม 2564].

ประวัติผู้จัดทำโครงการ



ชื่อ-สกุล

นายเนติธร ปรีชาพัฒนานนท์

วัน เดือน ปีเกิด

12 มกราคม พ.ศ. 2542

ประวัติการศึกษา

ระดับมัธยมศึกษา

ระดับมัธยมศึกษาตอนปลาย พ.ศ. 2559 โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า

ระดับอุดมศึกษา

คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมไฟฟ้า พ.ศ. 2563
สถาบันเทคโนโลยีไทย-ญี่ปุ่น