



ระบบติดตามเอกสารและกรณีศึกษา โครงการจัดตั้งวิทยาศาสตร์ดิจิทัล  
Document tracking system and studies case, Project to establish the  
College of digital Science

จรัญญา ธนะทวี

6210110459

โครงการวิศวกรรมคอมพิวเตอร์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา  
วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์  
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

2565



ระบบติดตามเอกสารและกรณีศึกษา โครงการจัดตั้งวิทยาศาสตร์ดิจิทัล  
Document tracking system and studies case, Project to establish  
the College of digital Science

จรัญญา ณะทวี

6210110459

โครงการวิศวกรรมคอมพิวเตอร์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา

ญา

วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสงขลานครินทร์

2565

ชื่อโครงการ ระบบติดตามเอกสารและกรณีศึกษา โครงการจัดตั้งวิทยาลัยวิทยาศาสตร์  
ดิจิทัล  
ผู้จัดทำ นางสาวจรรยา ธนะทวี รหัสนักศึกษา 6210110459  
สาขาวิชา วิศวกรรมคอมพิวเตอร์  
ปีการศึกษา 2565

---

อาจารย์ที่ปรึกษาโครงการ

คณะกรรมการสอบ

.....  
( รศ.ดร. แสงสุรีย์ วสุพงศ์อัยยะ )

.....  
( ผศ. ชัชชัย เอ็งฉ้วน )

.....  
( ผศ. สุชน แซ่ว่อง )

.....  
( รศ. ทศพร กลมกวิงส์ )

โครงการนี้เป็นส่วนหนึ่งของรายวิชาโครงการวิศวกรรมคอมพิวเตอร์ 1 และ 2  
ตามหลักสูตร ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์  
มหาวิทยาลัยสงขลานครินทร์

.....  
( รศ.ดร.พิชญ์ ตันฑิย์ )

หัวหน้าสาขาวิชาวิศวกรรมคอมพิวเตอร์

## หนังสือรับรองความเป็นเอกลักษณ์

ข้าพเจ้าผู้ลงนามทำยนี้ ขอรับรองว่ารายงานฉบับนี้เป็นรายงานที่มีความเป็นเอกลักษณ์ โดยที่ ข้าพเจ้ามิได้การคัดลอกมาจากที่ใด เนื้อหาในรายงานทั้งหมดถูกรวบรวมจากการพัฒนาในขั้นตอนต่างๆ ของการจัดทำโครงการ หากมีส่วนหนึ่งส่วนใดที่จำเป็นต้องนำข้อความจากผลงานของบุคคลอื่นใดที่ไม่ใช่ตัวข้าพเจ้า ข้าพเจ้าได้ทำอ้างอิงถึงเอกสารเหล่านั้นไว้อย่างเหมาะสม และขอรับรองว่ารายงาน ฉบับนี้ไม่เคยเสนอต่อสถาบันใดมาก่อน

ผู้จัดทำ

.....  
( นางสาวจรรยา ณะทวี )

ชื่อโครงงาน	ระบบติดตามเอกสารและกรณีศึกษา โครงการจัดตั้งวิทยาลัยวิทยาศาสตร์ ดิจิทัล	
ผู้จัดทำ	นางสาวจรรยา ธนะทวี	รหัสนักศึกษา 6210110459
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	
ปีการศึกษา	2565	

### บทคัดย่อ

โครงงาน "ระบบติดตามเอกสารและกรณีศึกษา โครงการจัดตั้งวิทยาลัยวิทยาศาสตร์ดิจิทัล" มีจุดประสงค์เพื่อในการติดตามเอกสารของโครงการจัดตั้งฯ เนื่องจากปัญหาที่เกิดจากการหลักสูตรการเรียนของโครงการจัดตั้งฯ เป็นหลักสูตรภายใต้มาตรฐาน AUN-QA ซึ่งอยู่ในรูปแบบออนไลน์ส่งผลให้การส่งเอกสารได้นั้นยากจะติดตาม เนื่องจากมีการส่งออนไลน์ได้หลายช่องทาง ทำให้เอกสารที่ต้องส่งนั้นมีความไม่เป็นระเบียบสำหรับผู้รวบรวม รวมทั้งหน่วยงานของโครงการจัดตั้งฯ นี้ ไม่มีสถานที่หรือเจ้าหน้าที่ประจำในหน่วยงาน จึงส่งผลให้เมื่อมีเจ้าหน้าที่คนใหม่มา จะทำให้การทำงานของหน่วยงานมีความล่าช้าในการติดตามเอกสารมากขึ้น เพราะเนื่องจากต้องค้นหาเอกสารเก่าที่ไม่มีความเป็นระเบียบและแหล่งรวบรวมเอกสาร ดังนั้นผู้พัฒนาจึงมีแนวคิดในการสร้างระบบสื่อกลางที่สามารถอัปโหลดเอกสารและติดตามตรวจสอบสถานะเอกสารได้ โดยนักศึกษาสามารถล็อกอินเข้าสู่ระบบ เพื่อใช้งานฟังก์ชันการอัปโหลดเอกสาร การติดตามเอกสาร โดยการติดตามเอกสารจะมีสถานะของเอกสารแสดงขึ้นเพื่อให้นักศึกษาตรวจสอบว่าเอกสารของนักศึกษามีความเรียบร้อยหรือไม่ และเจ้าหน้าที่สามารถล็อกอินเข้าสู่ระบบ เพื่อใช้งานฟังก์ชันการตรวจสอบเอกสารที่นักศึกษาส่งมาและการอัปเดตเอกสารที่ได้ดำเนิน เพื่อให้ นักศึกษาได้ตรวจสอบว่าเอกสารมีความเรียบร้อย ระบบโดยใช้เฟรมเวิร์ค reactJS และ react-bootstrap เพื่อรองรับการแสดงผลได้ในหลายแพลตฟอร์ม

Project	Document tracking system and studies case, Project to establish the College of digital Science	
Author	Ms. Jarunya Thanatavee	ID 6210110459
Major Program	Computer Engineering	
Academic Year	2022	

### Abstract

Project "Document Tracking System and Case Studies The establishment of the Digital Science College" aims to track the documents of the establishment project. Due to problems arising from the study curriculum of the establishment project It is a course under the AUN-QA standard, which is in an online format, making the submission of documents difficult to track. because there are many online delivery channels This makes the documents to be submitted unorganized for the collectors. Including the agency of this establishment project, there is no place or permanent staff in the agency. As a result, when a new officer came It will cause the work of the agency to delay in tracking more documents. Because having to search for old documents that are not organized and collecting documents Therefore, developers have the idea of creating an intermediary system that can upload documents and monitor their status. where students can log in to the system To use the document upload function document tracking By tracking documents, the status of the document will be displayed for students to check whether their documents are in order or not. and staff can log in to the system To use the function of checking documents submitted by students and updating documents that have been carried out. so that students can check that the documents are complete system using reactJS framework and react-bootstrap to support display across multiple platforms

## กิตติกรรมประกาศ

โครงการงาน "ระบบติดตามเอกสารและกรณีศึกษา โครงการจัดตั้งวิทยาลัย  
วิทยาศาสตร์ดิจิทัล" สำเร็จลุล่วงมาได้ตามวัตถุประสงค์นั้นได้รับการสนับสนุนและการ  
ช่วยเหลือจากอาจารย์แสงสุรีย์ วสุพงศ์อัยยะ ที่คอยให้คำแนะนำชี้แนะในการแก้ไขปรับปรุง  
ข้อบกพร่องและอุปสรรคต่าง มาโดยตลอดจนส่งผลให้โครงการฉบับนี้สำเร็จตามวัตถุประสงค์  
ที่กำหนดไว้ จึงขอกราบขอบคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบคุณ ผศ.รัชชัย เอ็งฉ้วน ผศ.สุชน แซ่ว่องและรศ.ทศพร กมลวิวงศ์  
คณะกรรมการสอบทุกๆ ท่านที่ให้คำแนะนำและข้อเสนอแนะ ตลอดจนรุ่นพี่และเพื่อนๆ ทุก  
คนที่ได้ให้ความช่วยเหลือด้วยดีมาโดยตลอด

ชื่อผู้จัดทำ

.....

( จริญญา ณะทวี )

## สารบัญ

บทคัดย่อ .....	4
Abstract .....	5
กิตติกรรมประกาศ .....	6
สารบัญ .....	7
รายการภาพประกอบ .....	9
<b>บทที่ 1 บทนำ .....</b>	<b>11</b>
1.1 ความสำคัญและที่มาของโครงการ .....	11
1.2 วัตถุประสงค์ของโครงการ .....	12
1.3 ประโยชน์ที่คาดว่าจะได้รับ .....	12
1.4 ขอบเขตของโครงการ .....	12
<b>บทที่ 2 ทฤษฎีและหลักการ .....</b>	<b>13</b>
2.1 JWT .....	13
2.2 JavaScript .....	14
2.3 ฐานข้อมูล .....	16
2.3.1 Atomicity .....	17
2.3.2 Consistency .....	17
2.3.3 Isolation .....	17
2.3.4 Durability .....	17
2.4 เครื่องมือและเทคโนโลยีที่ใช้ .....	18
2.4.1 ReactJS .....	18
2.4.2 NodeJS .....	18
2.4.3 MySQL Workbench .....	18
<b>บทที่ 3 วิธีดำเนินงาน .....</b>	<b>19</b>
3.1 ภาพรวมของโครงการ .....	19
3.2 แนวคิดในการออกแบบระบบ .....	19
3.3 ส่วนประกอบของโครงการ .....	20
3.3.1 ขอบเขตฟังก์ชันการทำงานของผู้ใช้งานภายในระบบ .....	20
3.3.2 การออกแบบหน้าเว็บแอปพลิเคชัน .....	21
3.3.3 โครงสร้างของฐานข้อมูล .....	26
<b>บทที่ 4 ผลและวิเคราะห์ผล .....</b>	<b>28</b>



4.1	การออกแบบโครงสร้างเว็บแอปพลิเคชัน.....	28
4.1.1	รายละเอียดการพัฒนาเว็บแอปพลิเคชัน .....	28
4.2	การออกแบบโครงสร้างข้อมูล.....	33
4.2.1	โครงสร้างฐานข้อมูล.....	33
4.2.2	การออกแบบโดยใช้ REST API .....	35
<b>บทที่ 5</b>	<b>สรุปและข้อเสนอแนะ .....</b>	<b>40</b>
5.1	สรุป .....	40
5.2	ข้อเสนอแนะ .....	40
<b>บรรณานุกรม.....</b>		<b>41</b>
<b>ภาคผนวก .....</b>		<b>43</b>

## รายการภาพประกอบ

รูปที่ 1 โครงสร้างหน้าตาของเจ้า JWT [13].....	13
รูปที่ 2 การรองรับในอุปกรณ์ฮาร์ดแวร์ของภาษา JavaScript [4] .....	15
รูปที่ 3 แพ็คเกจสำเร็จรูปที่มาช่วยในการเขียนโปรแกรม [4].....	16
รูปที่ 4 ภาพรวมการทำงานของระบบ .....	19
รูปที่ 5 Usecase ของระบบ .....	20
รูปที่ 6 ฟังก์ชันการลงทะเบียน.....	21
รูปที่ 7 ฟังก์ชันเข้าสู่ระบบ .....	22
รูปที่ 8 ฟังก์ชันหน้าหลักของนักศึกษา .....	22
รูปที่ 9 ฟังก์ชันหน้าแรกของเจ้าหน้าที่ .....	23
รูปที่ 10 ฟังก์ชันหน้าแรกของอาจารย์ .....	23
รูปที่ 11 ฟังก์ชันอัปโหลดเอกสาร .....	24
รูปที่ 12 ฟังก์ชันสถานะเอกสาร .....	24
รูปที่ 13 ฟังก์ชันแสดงรายละเอียดของเจ้าหน้าที่ .....	25
รูปที่ 14 ฟังก์ชันแสดงรายละเอียดของนักศึกษาและอาจารย์.....	25
รูปที่ 15 ฟังก์ชันอัปเดตเอกสาร.....	26
รูปที่ 16 การแสดงโครงสร้างของฐานข้อมูล .....	27
รูปที่ 17 คำสั่งสร้างโปรเจกต์ .....	28
รูปที่ 18 คำสั่งการติดตั้ง react-bootstrap, bootstrap.....	28
รูปที่ 19 คำสั่งการติดตั้ง MUI.....	28
รูปที่ 20 หน้าลงทะเบียนผู้ใช้งานบนเดสก์ท็อป .....	29
รูปที่ 21 หน้าล็อกอินเมื่อมีล็อกอินสำเร็จ .....	29
รูปที่ 22 ฟังก์ชันหน้าหลักของเว็บแอปพลิเคชันบนเดสก์ท็อป .....	30
รูปที่ 23 ฟังก์ชันหน้าอัปโหลดเอกสารของนักศึกษา.....	31
รูปที่ 24 กำหนดค่าให้กับตัวแปร statusf .....	31
รูปที่ 25 ฟังก์ชันสถานะเอกสาร .....	31
รูปที่ 26 การกำหนดตัวแปรเพื่อให้เรียกใช้ข้อมูลจากฐานข้อมูล .....	32
รูปที่ 27 สถานะเอกสารเมื่ออัปโหลดเอกสาร.....	32
รูปที่ 28 สถานะเอกสารเมื่อเจ้าหน้าที่อัปเดตเอกสาร .....	32
รูปที่ 29 การติดตั้ง react-pdf-viewer ในรูปแบบ default-layout .....	32
รูปที่ 30 องค์ประกอบของ default-layout.....	33

รูปที่ 31 ฟังก์ชันดูรายละเอียดเอกสาร .....	33
รูปที่ 32 ฐานข้อมูลของเว็บแอปพลิเคชันระบบติดตามเอกสาร .....	33
รูปที่ 33 ฐานข้อมูลของหน้าลงทะเบียนและหน้าล็อกอิน .....	34
รูปที่ 34 ตัวอย่างการเก็บข้อมูลของหน้าลงทะเบียนและหน้าล็อกอิน .....	34
รูปที่ 35 ฐานข้อมูลของหน้าหลัก .....	34
รูปที่ 36 ฐานข้อมูลของสถานะเอกสาร .....	34
รูปที่ 37 ตัวอย่างโค้ดที่ใช้สร้างเชื่อมต่อฐานข้อมูล เพื่อจะทำการส่งและรับ API.....	35
รูปที่ 38 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันลงทะเบียน .....	35
รูปที่ 39 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันล็อกอิน .....	36
รูปที่ 40 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันอัปโหลดไฟล์ .....	36
รูปที่ 41 ตัวอย่างโค้ดของ upload.single .....	37
รูปที่ 42 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันหน้าหลัก .....	37
รูปที่ 43 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันการลบข้อมูลเอกสาร .....	38
รูปที่ 44 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันดูสถานะเอกสาร .....	38
รูปที่ 45 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันอัปเดตเอกสาร .....	39
รูปที่ 46 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันดูรายละเอียดเอกสาร .....	39

## บทที่ 1

### บทนำ

#### 1.1 ความสำคัญและที่มาของโครงการงาน

โครงการจัดตั้งวิทยาลัยวิทยาศาสตร์ดิจิทัล มหาวิทยาลัยสงขลานครินทร์ เป็นหน่วยงานนำร่องการใช้การบริหารงานแบบเสมือนได้รับการอนุมัติจัดตั้งโดยอธิการบดีมหาวิทยาลัยสงขลานครินทร์เมื่อวันที่ 10 มิถุนายน 2563 มีฐานะเทียบเท่าหน่วยงานระดับคณะ [1] ซึ่งโครงการจัดตั้งฯ มีการบริหารงานหลักสูตรระดับบัณฑิตศึกษาจำนวน 2 หลักสูตร ได้แก่ หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาการข้อมูล และหลักสูตรปรัชญาดุษฎีบัณฑิต สาขาวิทยาการข้อมูล สิ่งที่แตกต่างของการบริหารงานโครงการจัดตั้งฯ คือ การไม่มีบุคลากรประจำมีเพียงลูกจ้างโครงการเพื่อดำเนินงานด้านเอกสาร และมีคณะกรรมการจำนวน 2 ชุด ได้แก่ คณะกรรมการอำนวยการและคณะกรรมการประจำโครงการจัดตั้งวิทยาลัยวิทยาศาสตร์ดิจิทัล มีบุคลากรที่เกี่ยวข้องในฐานะคณะกรรมการ คณาจารย์ ผู้สอน คณาจารย์ ประจำหลักสูตรจากทั้ง 5 วิทยาเขตของมหาวิทยาลัยสงขลานครินทร์ ปัจจุบันโครงการจัดตั้งฯ มีนักศึกษาจำนวน 35 คน กระจายอยู่ทุกวิทยาเขต

โครงการจัดตั้งฯ มีกิจกรรมการจัดการเรียนการสอนตามแนวทางการบริหารหลักสูตรภายใต้มาตรฐาน AUN-QA มีการจัดกิจกรรมติดตามการเรียนการสอน การวิจัยของนักศึกษาในหลักสูตร โดยปัจจุบันการสื่อสารหลักได้แก่ email, website [1], Facebook group [2] และกลุ่ม line ส่งผลให้เกิดประเด็นปัญหาการรับทราบข้อมูล การสื่อสารข้อมูล และการติดตามข้อมูลในบางกรณี ไม่ว่าจะเป็นการติดตามคำร้อง การติดตามเอกสารการลงทะเบียน การติดตามเอกสารการศึกษาอื่น ๆ เนื่องจากประเด็นปัญหาของสถานที่ การไม่มีสถานที่หรือเจ้าหน้าที่ประจำในแต่ละวิทยาเขต และการไม่มีระบบสื่อสารกลางที่สามารถติดตามตรวจสอบสถานะเอกสารต่าง ๆ ทางผู้พัฒนาจึงมีแนวคิดในการสร้างระบบเพื่อช่วยในการติดตามเอกสารของกิจกรรมต่าง ๆ ในโครงการจัดตั้งวิทยาลัยวิทยาศาสตร์ดิจิทัล เพื่อใช้เป็นช่องทางกลางในการสื่อสารกับนักศึกษาและคณาจารย์ในโครงการ โดยเน้นการพัฒนา web application ที่มีคุณสมบัติ Responsive ให้รองรับการใช้งานผ่านอุปกรณ์ที่หลากหลาย โดยมีความสามารถในการรับเอกสาร รับเรื่องร้องเรียน แสดงสถานะของเอกสารผ่านหน้าเว็บ เมื่อพัฒนาแล้วเสร็จจะสามารถนำไปแขวนบนหน้าเว็บหลักของโครงการจัดตั้งฯ ต่อไป

## 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อออกแบบและพัฒนาระบบติดตามสถานะเอกสาร โครงการจัดตั้งวิทยาลัยวิทยาศาสตร์ดิจิทัล มหาวิทยาลัยสงขลานครินทร์
2. ทดสอบและสรุปผลการทดสอบระบบติดตามสถานะเอกสารที่พัฒนาขึ้น

## 1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ระบบติดตามสถานะเอกสารของโครงการจัดตั้งวิทยาลัยวิทยาศาสตร์ดิจิทัล มหาวิทยาลัยสงขลานครินทร์
2. นักศึกษาและคณาจารย์ภายใต้หลักสูตรของโครงการจัดตั้งวิทยาลัยวิทยาศาสตร์ดิจิทัล มีระบบเพื่อติดตามสถานะของเอกสารภายใน

## 1.4 ขอบเขตของโครงการ

1. นักศึกษาสามารถโหลดเอกสารเข้าระบบ พร้อมติดตามสถานะของเอกสารผ่านระบบได้
2. เจ้าหน้าที่โครงการสามารถอัปเดตสถานะของเอกสารและเรียกดูรายงานได้
3. ผู้บริหารสามารถตรวจสอบสถานะของเอกสารและเรียกดูรายงานได้
4. ระบบรองรับการทำงานบนอุปกรณ์ที่หลากหลาย

## บทที่ 2

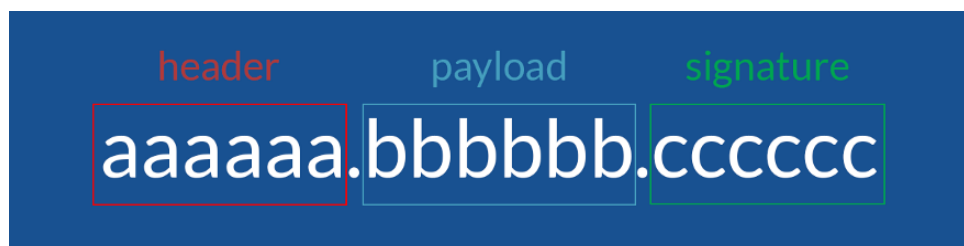
### ทฤษฎีและหลักการ

จากการศึกษาข้อมูลที่เกี่ยวข้องกับการพัฒนาเว็บแอปพลิเคชันสำหรับระบบติดตามเอกสารได้นำทฤษฎีหรือหลักการที่เกี่ยวข้องได้แก่ การยืนยันตัวตนโดยใช้ JWT การทำเว็บแอปพลิเคชันด้วยภาษา JavaScript และการจัดการระบบฐานข้อมูลที่ใช้ในการพัฒนา ซึ่งมีรายละเอียดดังนี้

#### 2.1 JWT

JSON Web Token (JWT) เป็นมาตรฐานเปิด (RFC 7519) ที่เข้ามาแก้ปัญหาการส่งข้อมูลอย่างปลอดภัยระหว่างกัน โดยที่ถูกออกแบบไว้ว่าจะต้องมีขนาดที่กระทัดรัด (Compact) และเก็บข้อมูลภายในตัว (Self-contained) ซึ่ง JWT จะเป็นชุดตัวอักษรชุดหนึ่ง โดยแบ่งออกเป็น 3 ท่อน ได้แก่ [13]

1. Header : ไว้เก็บว่าชุดข้อความนี้เข้ารหัสแบบใด [13]
2. Payload : เก็บข้อมูลจริง เช่น role ของผู้ใช้งาน [13]
3. Signature : ไว้ตรวจสอบเป็น Token ที่ถูกสร้างอย่างถูกต้องหรือไม่ เพราะหากมีแฮกเกอร์แฮกจะทำให้ Token เปลี่ยนแปลงไปและไม่สามารถเชื่อถือได้ [13]



รูปที่ 1 โครงสร้างหน้าตาของเจ้า JWT [13]

ซึ่ง JWT เข้ามาแก้ปัญหาดังนี้

ประสิทธิภาพ (Performance) : เนื่องจากความพิเศษของมันที่มีการเก็บข้อมูลในตัว (Self-contained) เลยทำให้มันลดการทำงานซ้ำๆ ไป ซึ่งข้อมูลเหล่านี้ไม่ได้อันตรายและเสี่ยงต่อความปลอดภัย เราเลยสามารถเก็บลง Payload ได้ ลดการทำงาน เพิ่มประสิทธิภาพ ไม่ต้อง Query หา Database ซ้ำๆ [13]

การขยายตัว (Scalability) :แม้จะมีการขยาย SERVER เพื่อในการรองรับการทำงาน JWT ก็จะสามารถทำงานได้เนื่องจากไม่มีการเก็บ token ไว้บน SERVER [13]

CORS (Cross-origin Resource Sharing) : จะทำเว็บไซต์หลายเว็บก็ไม่ต้องกลัวเรื่อง CORS เพราะสามารถทำการ Authentication ส่ง Token นี้ไป Verify Signature หากทุกอย่างถูกต้องก็นำไปใช้ได้เลย [13]

Cross-Site Request Forgery (CSRF, XSRF) :Token ไม่ได้เก็บไว้ใน Cookies แต่แนบไปกับ Header หรือ GET / POST Parameters ทำให้ไม่มีโอกาสเกิดการโจมตีแบบ CSRF [13]

## 2.2 JavaScript

JavaScript คือ Scripting Language จัดอยู่ในกลุ่มภาษาการเขียนโปรแกรมใช้ทั้งฝั่ง client-side และ server-side ในการสร้าง web pages แบบ interactive [3]

JavaScript นั้นถูกกำหนดมาตรฐานโดย European Computer Manufacturers Association (ECMA) เป็นองค์กรที่คอยพัฒนามาตรฐานและรายงานทางเทคนิคเพื่ออำนวยความสะดวกและสร้างมาตรฐานการใช้เทคโนโลยีการสื่อสารข้อมูลและอุปกรณ์อิเล็กทรอนิกส์ พร้อมส่งเสริมการใช้มาตรฐานอย่างถูกต้อง ซึ่ง ECMA ได้มีการพัฒนา JavaScript หลายเวอร์ชัน ได้แก่

JavaScript Edition 3 (เผยแผ่ปี 1999) \*\* ไม่มีเวอร์ชัน 4 และกลายเป็นพีเจอาร์ใน ES6 \*\*

JavaScript Edition 5 (เผยแผ่ปี 2009)

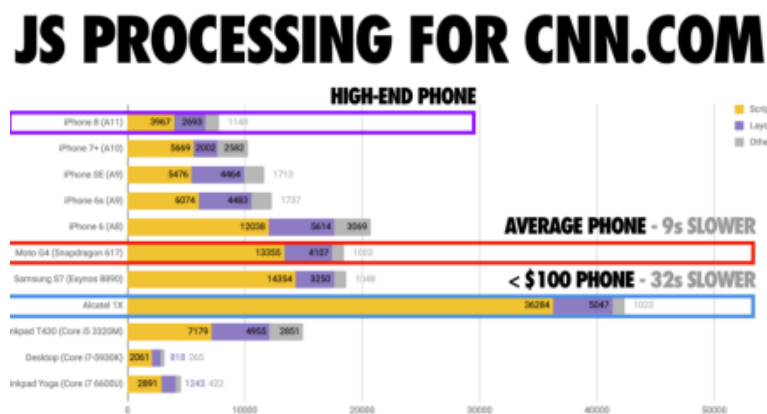
JavaScript Edition 6 = ECMAScript 2015 = ES2015 = ES6 (เผยแผ่ปี 2015)

JavaScript Edition 7 = ECMAScript 2016 = ES2016 = ES7 (เผยแผ่ปี 2016)

JavaScript Edition 8 = ECMAScript 2017 = ES2017 = ES8 (เผยแผ่ปี 2017)

JavaScript Edition 9 = ECMAScript 2018 = ES2018 = ES9 (เผยแผ่ปี 2018)

JavaScript มีการรองรับได้หลายพื้นที่การใช้งาน ไม่ว่าจะเป็น Web Browser เช่น Google Chrome, Firefox, Safari การนำมาใช้งานกับ Operating System เช่น Windows, Linux, Mac ด้วย NodeJS การนำไปใช้งานกับ Microcontroller เช่น ESP8266, NodeMCU ด้วย Espruino ข้อจำกัดของ JavaScript นั้นเน้นในด้านการรองรับ เนื่องจากการทำงานของภาษาเขียนโปรแกรมควรบอกถึงการรองรับอุปกรณ์หรือพื้นที่การใช้งานเพื่อพิจารณาของการนำไปใช้งาน ซึ่ง JavaScript จะไม่รองรับทุกพีเจอรืในทุกเบราว์เซอร์หรือ NodeJS เนื่องจาก JavaScript มีข้อจำกัดกับโปรแกรมเก่าอยู่ ได้มีการปรับปรุงและมีวิธีการแก้ไขแล้วแต่ก็ยังไม่ดีเท่ากับเวอร์ชันที่รองรับ ข้อจำกัดอีกส่วนคือฮาร์ดแวร์ที่ใช้งาน การจะเขียนโปรแกรมหนึ่งจะต้องคำนวณจำนวนผู้ใช้ อุปกรณ์ที่ใช้งานตามรูปที่ 5 เพื่อให้จัดการและออกแบบการเขียนโปรแกรมได้ และข้อจำกัดที่สำคัญอีกส่วนหนึ่งคือขนาดของไฟล์ การทำเว็บอาจจะส่งไฟล์มีขนาดใหญ่ได้เนื่องจากการตกแต่งความสวยงามโดยใช้แพ็คเกจสำเร็จรูปตามรูปที่ 6 มาช่วย อาจจะทำให้บางครั้งเกินความสามารถที่จำเป็นได้ จึงทำให้ไฟล์มีขนาดใหญ่มากจะส่งผลให้เว็บไซต์ที่โหลดได้ช้าขึ้นจากฝั่งผู้ใช้งาน [4]



รูปที่ 2 การรองรับในอุปกรณ์ฮาร์ดแวร์ของภาษา JavaScript [4]





รูปที่ 3 แพ็คเกจสำเร็จรูปที่มาช่วยในการเขียนโปรแกรม [4]

## 2.3 ฐานข้อมูล

ฐานข้อมูล คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ แฟ้มข้อมูล [6]

ระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกัน เข้าไว้ด้วยกันอย่างมีระบบมีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน [6] ซึ่งจะช่วยให้ผู้ใช้งานสามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยการนำซอฟต์แวร์มาช่วยในการทำงานระหว่างผู้ใช้กับโปรแกรมที่เกี่ยวข้องกับฐานข้อมูลเรียกว่าระบบจัดการการฐานข้อมูล (DBMS : Database Management System) ซึ่งระบบการจัดการฐานข้อมูลที่นำมาใช้กับโครงการนี้เป็นระบบการจัดการฐานข้อมูลแบบเชิงสัมพันธ์ (RDBMS) เนื่องจากฐานข้อมูลที่นำมาใช้มีความสัมพันธ์และเชื่อมโยงต่อกัน ซึ่งการนำระบบจัดการฐานข้อมูลนี้มาใช้จะช่วยในการจัดเก็บข้อมูลและดึงข้อมูลจำนวนมากได้อย่างมีประสิทธิภาพของระบบร่วมกันและความสะดวกในการใช้งาน [7]

ระบบฐานข้อมูลแบบเชิงสัมพันธ์จะประกอบด้วยฟังก์ชัน ACID คือ หลักการที่ใช้ในการบริหารจัดการ database transaction เพื่อการันตีว่าทุกๆ คำสั่งที่เกิดขึ้นกับระบบฐานข้อมูลจะมี ความถูกต้องอยู่เสมอ ACID ย่อมาจาก Atomicity, Consistency, Isolation และ Durability [8]

### 2.3.1 Atomicity

ระบบฐานข้อมูลจะต้องสามารถยืนยันได้ว่าคำสั่งย่อยทุกๆ คำสั่งในชุดคำสั่งเดียวกัน จะต้องสมบูรณ์พร้อมกันทั้งหมดหรือถ้ามีคำสั่งใดผิดพลาดก็จะต้องยกเลิกคำสั่งอื่นๆ ในชุดคำสั่งเดียวกันด้วย [8]

### 2.3.2 Consistency

ความถูกต้องสอดคล้องกันของข้อมูลที่จะเกิดหลังจากในแต่ละคำสั่งทำงานเสร็จสมบูรณ์ ซึ่งหมายถึงในแต่ละคำสั่งที่เกิดขึ้นกับระบบฐานข้อมูล จะต้องเป็นคำสั่งที่ไม่กระทบกับข้อมูลที่มีอยู่แล้วในระบบฐานข้อมูลและผลลัพธ์ทั้งหมดที่ออกมาหลังจากทำงานตามคำสั่งนั้นไปแล้ว จะต้องยังคงความถูกต้องอยู่เสมอ [8]

### 2.3.3 Isolation

ความเป็นเอกเทศของทรานซิกชัน หมายถึง ในแต่ละคำสั่งที่กำลังทำงานอยู่และยังไม่เสร็จสมบูรณ์ (commit) คำสั่งนั้นจะต้องมีความเป็นเอกเทศต้องยังไม่มีผลกับทรานซิกชันอื่นๆ จนกระทั่งจะ commit คำสั่งนั้น [8]

### 2.3.4 Durability

คือการเก็บรักษาคำสั่งที่เสร็จสมบูรณ์แล้ว (committed transaction) ให้มีความคงทน ถาวร หมายถึง คำสั่งใดๆ ก็ตามที่ถูก commit แล้ว จะต้องถูกเก็บรักษาเพื่อให้ทำงานได้จนเสร็จ สมบูรณ์ ถึงแม้ว่าจะเกิดปัญหาขึ้นกับระบบในระหว่างการทำงานนั้น [8] ซึ่งประโยชน์ในการนำระบบจัดการฐานข้อมูลเชิงสัมพันธ์ จะช่วยในการดำเนินการข้อมูลได้ดี ปรับปรุงกระบวนการตัดสินใจ การใช้ตารางเพื่อจัดเก็บข้อมูลพร้อมทั้งยังเพิ่มความปลอดภัยของข้อมูลที่จัดเก็บไว้ในฐานข้อมูลและผู้ใช้สามารถปรับแต่งการเข้าถึงเพื่อจำกัดเนื้อหาที่มี อีกทั้งยังมีความยืดหยุ่นในการอัปเดตข้อมูลจะมีประสิทธิภาพมากขึ้น การบำรุงรักษาผู้ดูแลระบบฐานข้อมูลสามารถบำรุงรักษา ควบคุมและอัปเดตข้อมูลในฐานข้อมูลได้อย่างง่ายดาย และมีการสำรองข้อมูลได้อย่างง่าย เนื่องจากเครื่องมืออัตโนมัติที่รวมอยู่ในระบบจัดการฐานข้อมูลนี้และโครงสร้างข้อมูลรูปแบบตารางที่ใช้งานนั้นง่ายต่อการทำความเข้าใจและโครงสร้างที่รายการจะถูกจับคู่โดยการสืบค้นข้อมูล [7]

## 2.4 เครื่องมือและเทคโนโลยีที่ใช้

### 2.4.1 ReactJS

React คือ JavaScript Library อีกตัวหนึ่งที่มีหน้าที่สร้างงานทางด้าน Front-end และ Back-end :ซึ่งลักษณะของ React จะแบ่งส่วนของการแสดงผลออกเป็นหลายๆ ส่วน จะเรียกว่า Component (คอมโพเนนท์) เช่น ส่วนของ header, footer เป็นต้น และเมื่อจะใช้งานก็จะนำ Component ต่างๆมาประกอบร่างกันและแสดงผลออกมาเป็นหน้าเว็บแอปพลิเคชัน[9]

### 2.4.2 NodeJS

Node JS คือการเขียนโปรแกรมฝั่งเซิร์ฟเวอร์ด้วยภาษา JavaScript เป็นเสมือน Platform ซึ่ง Node JS จะมีการประมวลผลที่เร็วจึงส่งผลให้แอปพลิเคชันที่ใช้นั้นสะดวกและรวดเร็ว [10] รวมทั้งยังเป็นโปรแกรมที่สามารถใช้งานทั้งบนระบบปฏิบัติการ Windows, Linux และ Mac OS X ดังนั้น Node JS นี้จึงเป็นการเขียนครั้งเดียวแต่ทำงานได้ทุกที่ ( write one, run anywhere ) [11]

### 2.4.3 MySQL Workbench

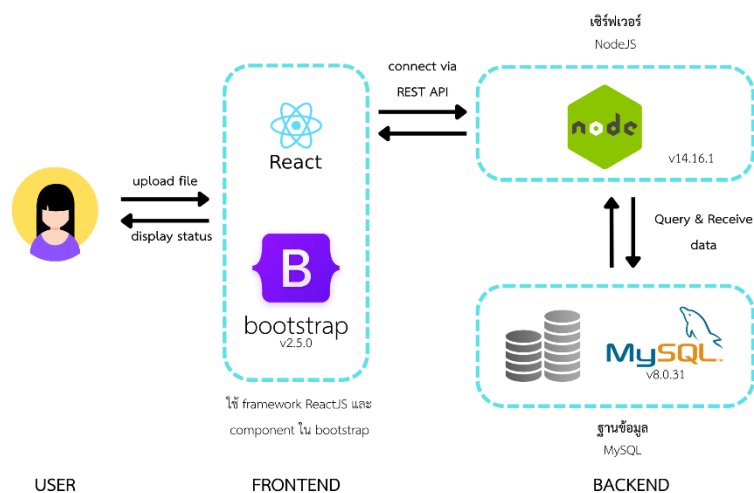
MySQL คือ โปรแกรมระบบจัดการฐานข้อมูลที่มีหน้าที่เก็บข้อมูลอย่างเป็นระบบรองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูลที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่นอย่างบูรณาการ เพื่อให้ได้ระบบงานที่รองรับความต้องการของผู้ใช้ เช่น ทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) MySQL เป็นโปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลายและเป็นระบบฐานข้อมูลโอเพนซอร์ส (Open Source) ที่ถูกนำไปใช้งาน ซึ่ง MySQL จึงถูกจัดให้นำมาใช้เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS : Relational Database Management System) [12]

## บทที่ 3

### วิธีดำเนินงาน

#### 3.1 ภาพรวมของโครงการ

ระบบติดตามเอกสาร เป็นเว็บแอปพลิเคชันเพื่ออำนวยความสะดวกให้สำหรับนักศึกษาและเจ้าหน้าที่ ซึ่งในด้าน Frontend จะพัฒนาด้วย ReactJS และ Bootstrap เพื่อให้มีรองรับการแสดงผลได้ในหลายแพลตฟอร์ม เช่น โทรศัพท์ แท็บเล็ตและโน้ตบุ๊ก และในด้าน Backend จะใช้ NodeJS เป็นตัวจัดการและ MySQL เป็นฐานข้อมูลที่จะเก็บข้อมูลต่างๆ ภายในเว็บพลิเคชันนี้ ส่วนการติดต่อ Frontend กับ Backend จะใช้ REST API ในการพัฒนา



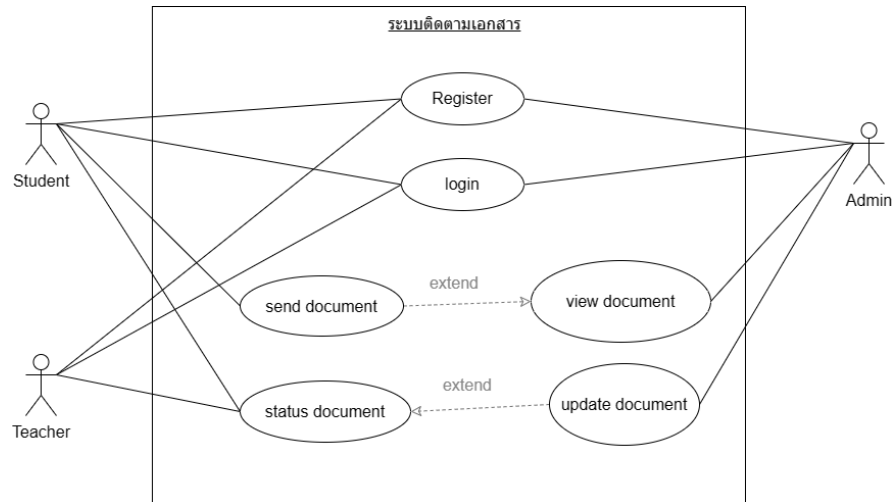
รูปที่ 4 ภาพรวมการทำงานของระบบ

#### 3.2 แนวคิดในการออกแบบระบบ

1. ผู้ใช้ที่เป็นนักศึกษาสามารถโหลดเอกสารเข้าระบบได้ โดยระบบรองรับเอกสาร .pdf นักศึกษาสามารถตรวจสอบสถานะของเอกสารตนเองได้
2. ผู้ใช้ที่เป็นเจ้าหน้าที่จะดึงเอกสารจากฐานข้อมูลมาตรวจสอบว่ามีเอกสารต้องส่งให้อาจารย์ท่านใด มีคำร้องเอกสารใดๆ พร้อมทั้งมีการปรับสถานะของเอกสารในระบบ
3. ผู้ใช้ที่เป็นผู้บริหารสามารถดูรายงานสถานะของเอกสารรายคน รายเดือนได้
4. ผู้ใช้ทุกประเภทสามารถดูตารางในช่วงที่เกี่ยวข้องได้

### 3.3 ส่วนประกอบของโครงงาน

#### 3.3.1 ขอบเขตฟังก์ชันการทำงานของผู้ใช้งานภายในระบบ



รูปที่ 5 Usecase ของระบบ

ระบบติดตามเอกสารและกรณีศึกษา มีผู้ใช้หลักๆ 3 ประเภท คือ นักศึกษา อาจารย์ และเจ้าหน้าที่ ประเภทของผู้ใช้งานมีความจำเป็นต้องมีการออกแบบให้มีความชัดเจนถึงการทำงานจากรูปที่ 8 แสดงให้เห็นดังนี้

- นักศึกษา

นักศึกษาต้องมีการลงทะเบียนผู้ใช้งานเพื่อเข้าสู่ระบบ เมื่อล็อกอินเข้าสู่ระบบ จะสามารถส่งเอกสารที่นักศึกษาต้องการจะแจ้งร้องขอ สามารถดูรายละเอียดของเอกสารพร้อมทั้งลบคำร้องขอเอกสารได้ และนักศึกษาติดตามสถานะของเอกสารได้เมื่อมีเอกสารมีความสำเร็จ นักศึกษาสามารถโหลดเอกสารได้

- อาจารย์

อาจารย์ต้องมีการลงทะเบียนผู้ใช้งานแล้วล็อกอินเข้าสู่ระบบ สามารถดูเอกสารที่นักศึกษาส่งและสถานะของเอกสารได้ พร้อมทั้งสามารถค้นหารายชื่อ ค้นหาหัวข้อเอกสาร และสามารถดูแยกเป็นสถานะ แยกเดือนได้

- เจ้าหน้าที่

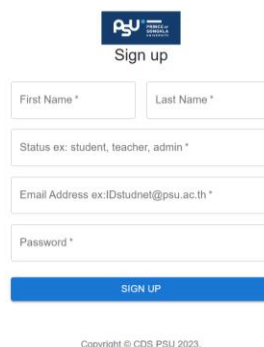
เจ้าหน้าที่ต้องมีการลงทะเบียนผู้ใช้งานแล้วเข้าสู่ระบบ เจ้าหน้าที่สามารถอัปเดตเอกสารได้ สามารถดูสถานะเอกสารพร้อมทั้งค้นหารายชื่อ หัวข้อเอกสาร แยกสถานะและแยกเป็นรายเดือนได้

### 3.3.2 การออกแบบหน้าเว็บแอปพลิเคชัน

หน้าเว็บแอปพลิเคชันสำหรับนักเรียน อาจารย์และเจ้าหน้าที่ จะเริ่มต้นหลังจากผู้ใช้งานลงทะเบียนและล็อกอินผ่านหน้าล็อกอิน เพื่อเข้าสู่หน้าหลัก โดยหน้าหลักจะแสดงตารางเอกสารที่นักศึกษาได้อัพโหลดและแถบเมนูสำหรับอัปโหลดเอกสารและดูสถานะเอกสาร

- การลงทะเบียน

ผู้ใช้งานต้องลงทะเบียนโดยประกอบไปด้วยชื่อ-นามสกุล สถานะในการศึกษา อีเมล และพาสเวิร์ดในการเข้าสู่ระบบ

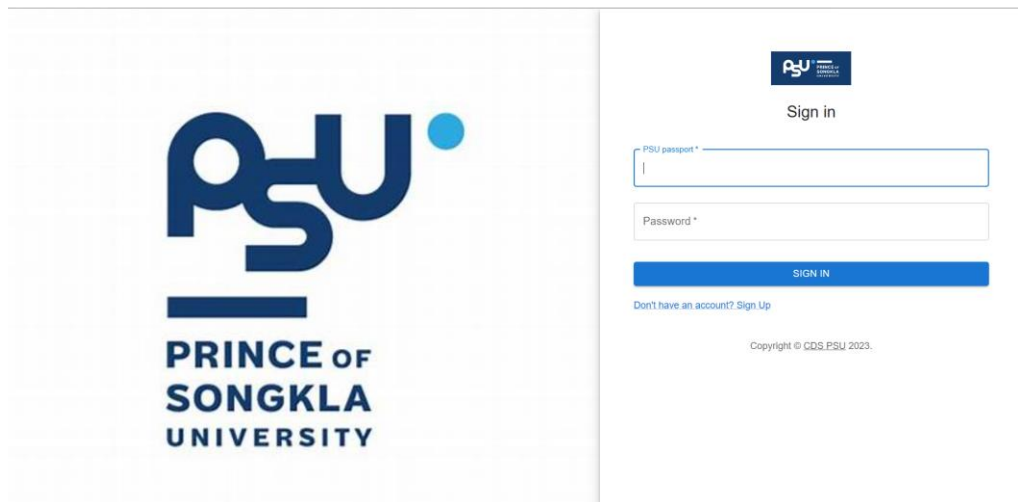


The image shows a 'Sign up' form for PSU. At the top is the PSU logo and the text 'Sign up'. Below this are five input fields: 'First Name \*', 'Last Name \*', 'Status ex: student, teacher, admin \*', 'Email Address ex: idstudnet@psu.ac.th \*', and 'Password \*'. A blue 'SIGN UP' button is at the bottom. Below the button is the copyright notice 'Copyright © CDS PSU 2023'.

### รูปที่ 6 ฟังก์ชันการลงทะเบียน

- การเข้าสู่ระบบ

ผู้ใช้งานลงทะเบียนเข้าสู่ระบบเรียบร้อยแล้ว ผู้ใช้งานก็จะนำอีเมลและพาสเวิร์ดที่ลงทะเบียนมาเข้าสู่ระบบ เพื่อใช้งานระบบ



รูปที่ 7 ฟังก์ชันเข้าสู่ระบบ

- หน้าหลัก

หน้าหลักแสดงตารางเอกสารที่นักศึกษาอัปโหลด ซึ่งนักศึกษาสามารถดูเอกสารที่อัปโหลดและลบเอกสารได้ ส่วนเจ้าหน้าที่จะสามารถดูเอกสารและอัปเดตเอกสารได้ และส่วนอาจารย์สามารถดูเอกสารที่นักศึกษาหรือเจ้าหน้าที่อัปเดตได้เท่านั้น ซึ่งจะถูกเก็บและดึงจากฐานข้อมูลโดยใช้ REST API ทั้งจะมีแถบเมนูเพื่อให้สามารถตรวจสอบสถานะเอกสารได้

Document System				Mark	Logout
Document Upload Projects With Mysql database				Add User	
#	Topic	Action	Status		
1	การยื่นเรื่องขอพบอธิการบดี	6210110XXX	Submitted		
2	การยื่นเอกสารจองหอ	6210110XXX	Submitted		
3	การยื่นคำร้องการเรียน	6210110XXX	Submitted		
4	การยื่นเรื่องขอพบอธิการบดี	6210110800	Submitted		

รูปที่ 8 ฟังก์ชันหน้าหลักของนักศึกษา

Document System

Upload

Status

admin

teacher

Mark

Logout

Document Upload Projects With Mysql database

Search

Search

#	Topic	name	Status		
1	การยื่นเรื่องขอหน่วยกิตเพิ่ม	6210110XXX	เอกสารยื่น		
2	การยื่นเอกสารร้องขอ	6210110XXX	เอกสารยื่น		
3	การยื่นคำร้องการเขียน	6210110XXX	เอกสารยื่น		
4	การยื่นเรื่องขอหน่วยกิตเพิ่ม	6210110800	เอกสารยื่น		

รูปที่ 9 ฟังก์ชันหน้าแรกของเจ้าหน้าที่

Document System

Upload

Status

admin

teacher

Mark

Logout

Document Upload Projects With Mysql database

#	Topic	name	Status	
1	การยื่นเรื่องขอหน่วยกิตเพิ่ม	6210110XXX	เอกสารยื่น	
2	การยื่นเอกสารร้องขอ	6210110XXX	เอกสารยื่น	
3	การยื่นคำร้องการเขียน	6210110XXX	เอกสารยื่น	
4	การยื่นเรื่องขอหน่วยกิตเพิ่ม	6210110800	เอกสารยื่น	

รูปที่ 10 ฟังก์ชันหน้าแรกของอาจารย์

- หน้าอัปโหลดเอกสาร

นักศึกษาต้องป้อนข้อมูลการอัปโหลดเอกสารโดยจะประกอบด้วย หัวข้อเอกสาร ชื่อไฟล์ ไฟล์เอกสารและหมายเหตุที่นักศึกษาจะแจ้ง ซึ่งการป้อนข้อมูลทั้งหมดจะถูกนำไปเก็บไว้ที่ฐานข้อมูลโดยใช้ REST API



Document System
Upload
Status
admin
teacher
Mark
Logout

### Upload Your File Here

Topic

FileName

Select Your File

เลือกไฟล์

บันทึกไฟล์ที่เลือก

Detail

Submit

รูปที่ 11 ฟังก์ชันอัปโหลดเอกสาร

- หน้าแสดงสถานะเอกสาร

แสดงเอกสารที่นักศึกษาอัปโหลดพร้อมกับสถานะเอกสารว่าอยู่ในขั้นตอนใด ซึ่งในการส่งเอกสารจะมีสถานะรอการดำเนิน แต่ถ้าเจ้าหน้าที่อัปเดตสถานะเอกสารก็จะเปลี่ยนเป็นเอกสารสำเร็จ ซึ่งการแสดงผลสถานะนั้นเกิดจากการดึงฐานข้อมูลจากการใช้ REST API ซึ่งการเปลี่ยนนี้เกิดการส่งค่าเมื่อนักศึกษาเรียกใช้หน้าอัปโหลดเอกสารและเปลี่ยนสถานะเมื่อเจ้าหน้าที่อัปเดตเอกสาร

Document System
Upload
Status
admin
teacher
Mark
Logout

### Status Your File Here

สถานะเอกสาร

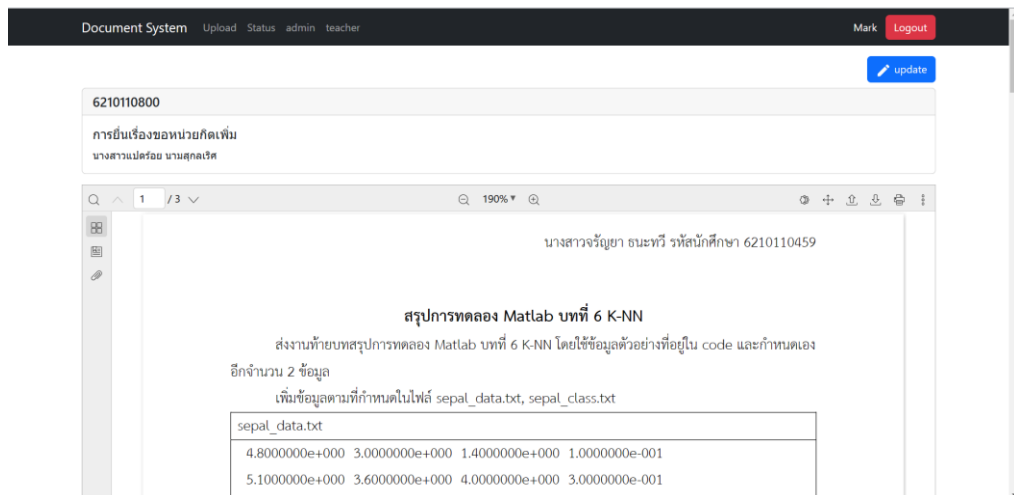
Search

#	Topic	name	status	
1	การยื่นเรื่องขอหน่วยกิตเต็ม	6210110XXX	เอกสารสำเร็จ	
2	การยื่นเอกสารร้องขอ	6210110XXX	เอกสารสำเร็จ	
3	การยื่นคำร้องการเรียน	6210110XXX	เอกสารสำเร็จ	
4	การยื่นเรื่องขอหน่วยกิตเต็ม	6210110800	เอกสารสำเร็จ	

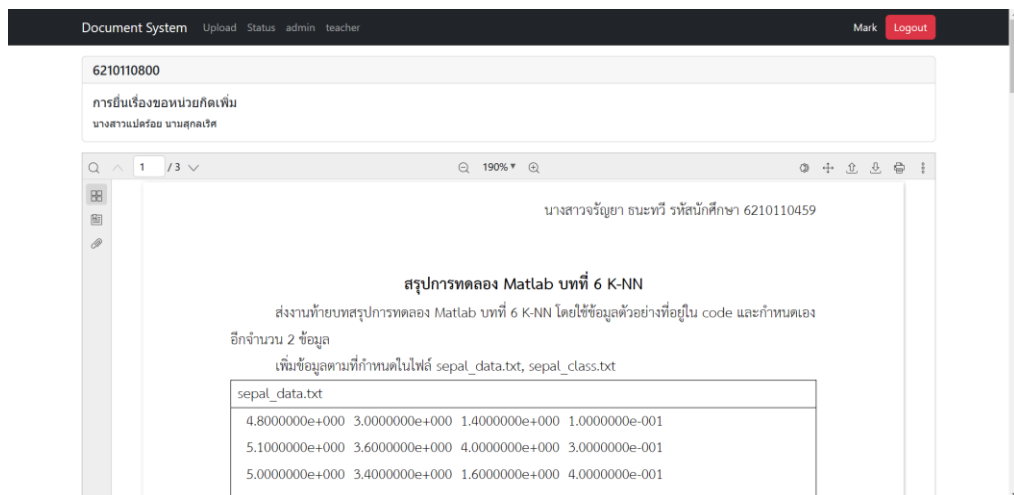
รูปที่ 12 ฟังก์ชันสถานะเอกสาร

- หน้าแสดงรายละเอียด

หน้าแสดงรายละเอียดจะประกอบด้วยชื่อไฟล์ หัวข้อเอกสาร หมายเหตุที่นักศึกษาแจ้งไว้ และไฟล์เอกสาร ซึ่งจะมีความแตกต่างกันของเจ้าหน้าที่และนักศึกษากับอาจารย์ที่เจ้าหน้าที่จะมีปุ่มเพื่อให้อัปเดตเอกสารที่เจ้าหน้าที่ดำเนินการแล้ว แต่นักศึกษาและอาจารย์จะสามารถดูและโหลดเอกสารได้เท่านั้น โดยการดูรายละเอียดนั้นจะเกิดจากรีเควส REST API ซึ่งจะมีการเรียกใช้ตัวแปร id เพื่อเปิดแสดงข้อมูลที่ละเอียดขึ้น ซึ่งผู้ใช้ทุกคนจะสามารถโหลดเอกสารในหน้าแสดงรายละเอียด



รูปที่ 13 ฟังก์ชันแสดงรายละเอียดของเจ้าหน้าที่



รูปที่ 14 ฟังก์ชันแสดงรายละเอียดของนักศึกษาและอาจารย์

- หน้าอัปเดตเอกสาร

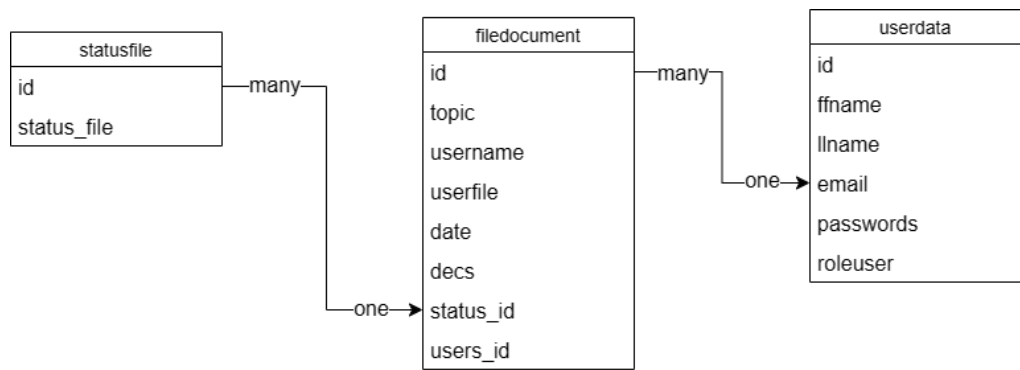
หน้าอัปเดตเอกสารจะมี 2 ส่วนหลักคือ ส่วนแสดงรายละเอียดของข้อมูลที่อัปเดตเอกสารนั้น ซึ่งจะประกอบด้วยชื่อไฟล์ หัวข้อเอกสารและหมายเหตุ ส่วนที่สองคือส่วนที่อัปเดตไฟล์เอกสารที่เจ้าหน้าที่จะอัปเดต ซึ่งทุกส่วนจะมีการเรียกใช้ REST API โดยส่วนแรกจะเรียกใช้ในการดึงฐานข้อมูลที่ละเอียดของฐานข้อมูลนั้น และอีกส่วนจะเรียกให้เป็นการอัปเดตข้อมูลลงฐานข้อมูล



รูปที่ 15 ฟังก์ชันอัปเดตเอกสาร

### 3.3.3 โครงสร้างของฐานข้อมูล

โครงสร้างของฐานข้อมูลเว็บแอปพลิเคชันระบบติดตามเอกสาร จะมีฐานข้อมูลหลักคือ ฐานข้อมูลไฟล์เอกสาร เช่น ชื่อหัวข้อเอกสาร ชื่อไฟล์ ไฟล์เอกสาร วันที่และหมายเหตุ รวมทั้งสถานะเอกสารที่เปลี่ยนแปลงตามการอัปเดตและชื่อผู้ใช้งานที่อัปเดตเอกสาร เนื่องจากฟังก์ชันต่างๆ ในเว็บแอปพลิเคชันระบบติดตามเอกสารต้องมีความสัมพันธ์ฐานข้อมูลที่กล่าวข้างต้นเพื่อดำเนินดึงและเก็บไว้ในฐานข้อมูลต่างๆ ได้อย่างมีระเบียบมากขึ้น แสดงได้ดังรูปที่ 19



รูปที่ 16 การแสดงโครงสร้างของฐานข้อมูล

## บทที่ 4

### ผลและวิเคราะห์ผล

เว็บแอปพลิเคชันพัฒนาด้วย ReactJS เป็น Front-End Framework และใช้ react-bootstrap, bootstrap, MUI เป็นการออกแบบองค์ประกอบต่างๆภายในเว็บแอปพลิเคชันนี้ ทำให้เว็บแอปพลิเคชันสามารถรองรับการแสดงผลได้ในหลายๆ อุปกรณ์ เช่น โทรศัพท์ คอมพิวเตอร์ แท็บเล็ต โน้ตบุ๊ก โดยมีรายละเอียดการพัฒนาดังนี้

#### 4.1 การออกแบบโครงสร้างเว็บแอปพลิเคชัน

- การสร้างโปรเจกต์ด้วยคำสั่งตามรูปที่ 17 รูปที่ 18 และรูปที่ 19 ใน Terminal

```
npx create-react-app my-app
```

รูปที่ 17 คำสั่งสร้างโปรเจกต์

```
npm install react-bootstrap bootstrap
```

รูปที่ 18 คำสั่งการติดตั้ง react-bootstrap, bootstrap

```
npm install @mui/material @emotion/react @emotion/styled
```

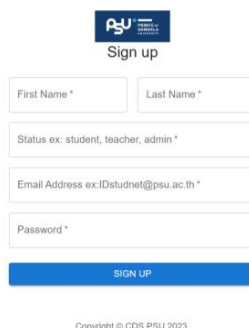
รูปที่ 19 คำสั่งการติดตั้ง MUI

##### 4.1.1 รายละเอียดการพัฒนาเว็บแอปพลิเคชัน

หน้าเว็บแอปพลิเคชันสำหรับนักเรียน อาจารย์และเจ้าหน้าที่ ฟังก์ชันที่พัฒนาแล้วจะประกอบไปด้วยหน้าลงทะเบียนเพื่อให้ผู้ใช้สามารถล็อกอินเข้าสู่ระบบได้ หน้าล็อกอินจะเป็นหน้าที่เข้าสู่หน้าหลักซึ่งหน้าหลักจะเชื่อมการใช้งานในฟังก์ชันต่างๆ ที่ประกอบด้วยฟังก์ชันหน้าหลัก ฟังก์ชันการอัปโหลดเอกสาร ฟังก์ชันสถานะเอกสาร ฟังก์ชันการอัปเดตเอกสารและฟังก์ชันแสดงรายละเอียด

#### 4.1.1.1 ฟังก์ชันลงทะเบียน

ใช้ text-field จาก MUI ในการรับค่าข้อมูลที่ใช้ป้อน ซึ่งประกอบด้วย First name, Last name, Status, Email และ Password และปุ่ม Sing up ใช้ Buttons จาก MUI

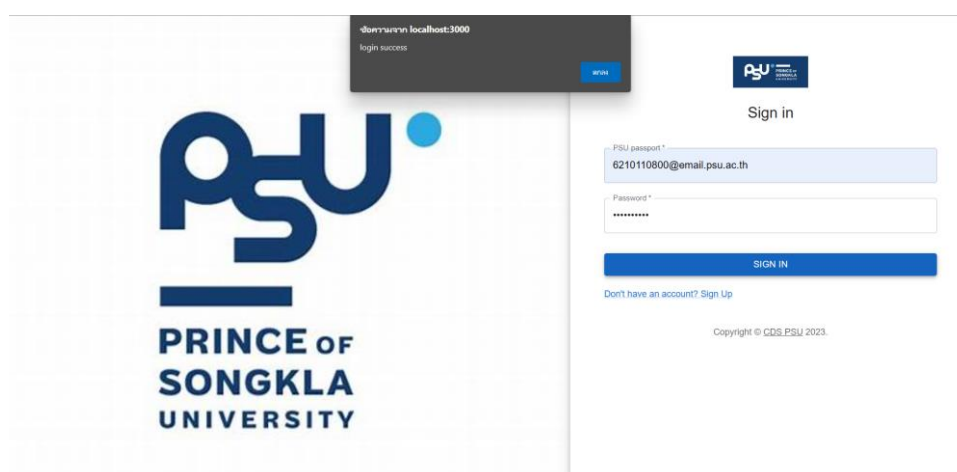


The image shows a 'Sign up' form for Prince of Songkla University (PSU). It features a blue header with the PSU logo and the text 'Sign up'. Below the header, there are five input fields: 'First Name \*', 'Last Name \*', 'Status ex: student, teacher, admin \*', 'Email Address ex: IDstudent@psu.ac.th \*', and 'Password \*'. A blue 'SIGN UP' button is at the bottom. The footer contains the text 'Copyright © CDS PSU 2023.'.

รูปที่ 20 หน้าลงทะเบียนผู้ใช้งานบนเดสก์ท็อป

#### 4.1.1.2 ฟังก์ชันล็อกอิน

ใช้ templates จาก MUI ซึ่งประกอบด้วย text-field ที่ใช้รับค่าจากผู้ใช้งานที่ป้อน นั่นคือ email และ password ปุ่ม Sign in ที่มาจาก button เพื่อใช้ในเรียก REST API มาตรวจสอบ email และ password ตรงตามที่ฐานข้อมูลหรือไม่ โดยถ้าการล็อกอินข้อมูลถูกต้องจะมี alert แสดงขึ้นมาพร้อมกับข้อความ” Login success”



The image shows a 'Sign in' form for Prince of Songkla University (PSU). It features a blue header with the PSU logo and the text 'Sign in'. Below the header, there are two input fields: 'PSU passport \*' (containing the email address 6210110800@email.psu.ac.th) and 'Password \*'. A blue 'SIGN IN' button is at the bottom. A dark grey alert box at the top left displays the text 'เชื่อมจาก localhost:3000' and 'login success'. Below the button, there is a link 'Don't have an account? Sign Up'. The footer contains the text 'Copyright © CDS PSU 2023.'.

รูปที่ 21 หน้าล็อกอินเมื่อมีล็อกอินสำเร็จ

#### 4.1.1.3 ฟังก์ชันหน้าหลัก

ใช้ Table จาก bootstrap ในการออกแบบตารางเอกสารเมื่อมีการส่งไฟล์เอกสารแล้ว โดยตารางจะประกอบไปด้วยชื่อหัวข้อเอกสาร ชื่อไฟล์ที่นักศึกษาตั้ง สถานะการส่งเอกสารสำเร็จ ปุ่มการดูรายละเอียดและปุ่มการลบโดยไอคอนปุ่มที่ใช้ได้มาจาก MUI ให้เกิดความสวยงามและมองง่าย ซึ่งข้อมูลที่แสดงขึ้นในหน้าหลักนี้เกิดการดึงข้อมูลจาก REST API ที่ได้จากหน้าอัปโหลดเอกสารที่เก็บไว้ในฐานข้อมูล

Document System

UploadStatusadminteacher

MarkLogout

Document Upload Projects With Mysql database

Add User

#	Topic	Action	Status		
1	การยื่นเรื่องขอหน่วยกิตเพิ่ม	6210110XXX	Submitted		
2	การยื่นเอกสารร้องขอ	6210110XXX	Submitted		
3	การยื่นคำร้องการเรียน	6210110XXX	Submitted		
4	การยื่นเรื่องขอหน่วยกิตเพิ่ม	6210110800	Submitted		

รูปที่ 22 ฟังก์ชันหน้าหลักของเว็บแอปพลิเคชันบนเดสก์ท็อป

#### 4.1.1.4 ฟังก์ชันอัปโหลดเอกสาร

จะใช้ Form ในการออกแบบรับค่าข้อมูลที่นักศึกษาป้อนเข้ามา โดยในแต่ละฟอร์มการใช้งานจะมีชนิดที่แตกต่างกัน โดยตัวแปรหัวข้อเอกสาร ชื่อไฟล์เอกสารและหมายเหตุจะรับข้อมูลเป็นรูปแบบ text ส่วนฟอร์มที่ใส่เอกสารจะมีการรับข้อมูลในรูปแบบไฟล์และมีการกำหนดไฟล์ว่ารับเฉพาะ PDF เท่านั้นด้วย

Document System Upload Status admin teacher Mark Logout

### Upload Your File Here

Topic  
topic

FileName  
filename

Select Your File  
เลือกไฟล์ โบบไฟล์ที่เลือก

Detail

Submit

รูปที่ 23 ฟังก์ชันหน้าอัปโหลดเอกสารของนักศึกษา

#### 4.1.1.5 ฟังก์ชันหน้าสถานะของเอกสาร

เป็นดึงข้อมูลที่ได้จากการเรียกใช้ REST API จากการกรอกข้อมูลที่หน้าอัปโหลดเอกสารลงไปฐานข้อมูล โดยหน้าสถานะที่มีการแสดงสถานะเอกสารซึ่งได้กำหนดตัวแปรสถานะเอกสารตั้งแต่เรียกกรอกข้อมูลในหน้าอัปโหลด เนื่องจากได้มีการบันทึกข้อมูลสถานะของเอกสารในฐานข้อมูลตามตัวอย่าง

```
const [statusf] = useState(1);
```

รูปที่ 24 กำหนดค่าให้กับตัวแปร statusf

Document System Upload Status admin teacher Mark Logout

### Status Your File Here

สถานะเอกสาร Search Search

#	Topic	name	status	
1	การยื่นเรื่องขอหน่วยกิตเพิ่ม	6210110XXXX	เสร็จสิ้น	
2	การยื่นเอกสารร้องขอ	6210110XXXX	เอกสารสำเร็จ	
3	การยื่นคำร้องการเรียน	6210110XXXX	เอกสารสำเร็จ	
4	การยื่นเรื่องขอหน่วยกิตเพิ่ม	6210110800	เอกสารสำเร็จ	

รูปที่ 25 ฟังก์ชันสถานะเอกสาร



#### 4.1.1.6 ฟังก์ชันอัปเดตเอกสาร

ใช้ Card จาก bootstrap แสดงผลข้อมูลที่เจ้าหน้าที่จะอัปเดตเอกสารโดยจะมีการเรียกใช้ REST API ที่มีตัวแปร id เป็นตัวกำหนดในการดึงข้อมูลและเมื่อเจ้าหน้าที่อัปเดตไฟล์เอกสารจะมีผลในการเปลี่ยนสถานะเอกสาร เนื่องจากการกำหนดในการอัปเดตสถานะเอกสารเมื่อเจ้าหน้าที่มีการอัปเดตไฟล์ตามตัวอย่างที่แสดง

```
var formData = new FormData();
formData.append("photo", file)
formData.append("status", 3)
```

รูปที่ 26 การกำหนดตัวแปรเพื่อให้เรียกใช้ข้อมูลจากฐานข้อมูล

5	การยื่นเรื่องขอหน่วยกิต	6210110888	รอการดำเนินการ	
---	-------------------------	------------	----------------	---

รูปที่ 27 สถานะเอกสารเมื่ออัปเดตเอกสาร

5	การยื่นเรื่องขอหน่วยกิต	6210110888	เอกสารสำเร็จ	
---	-------------------------	------------	--------------	---

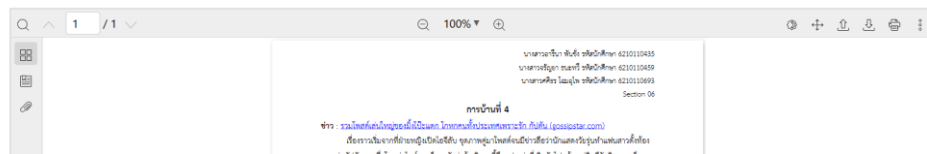
รูปที่ 28 สถานะเอกสารเมื่อเจ้าหน้าที่อัปเดตเอกสาร

#### 4.1.1.7 ฟังก์ชันดูรายละเอียดเอกสาร

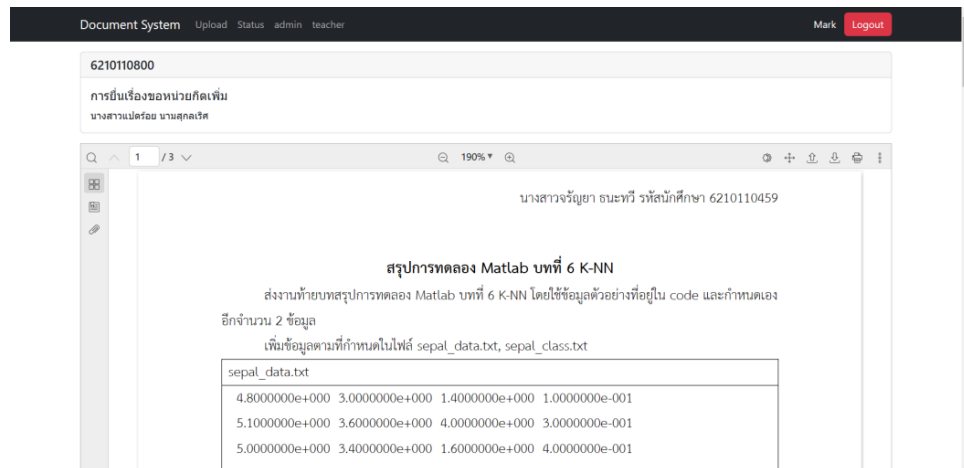
ใช้ react-pdf-viewer เป็นส่วนแสดงผลของข้อมูลไฟล์เอกสารที่นักศึกษาอัปเดต โดยรูปแบบการแสดงผลเป็นรูปแบบ default-layout ซึ่งจะมียอดประกอบการปรับเอกสาร การดาวน์โหลด การเปิดเอกสารเต็มหน้าและองค์ประกอบต่างๆ อีกมากมายตามตัวอย่าง และมีการแสดงผลข้อมูลในรูปแบบ Card จาก bootstrap ในส่วนของรายละเอียดข้อมูลที่เปิดดูในแต่ละ id ซึ่งเป็นตัวกำหนดการเรียกใช้ REST API ข้อมูลนั้นจากฐานข้อมูล

```
npm install '@react-pdf-viewer/default-layout';
```

รูปที่ 29 การติดตั้ง react-pdf-viewer ในรูปแบบ default-layout



รูปที่ 30 องค์ประกอบของ default-layout

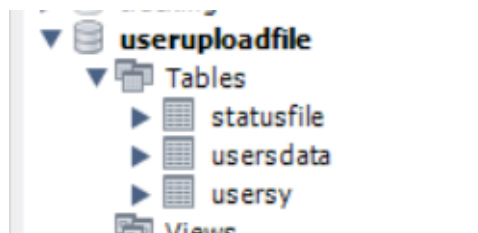


รูปที่ 31 ฟังก์ชันดูรายละเอียดเอกสาร

## 4.2 การออกแบบโครงสร้างข้อมูล

โครงงานนี้มีการออกแบบโครงสร้างฐานข้อมูลสำหรับ 7 ฟังก์ชันได้แก่ ฟังก์ชันลงทะเบียน ฟังก์ชันล็อกอิน ฟังก์ชันหน้าหลัก ฟังก์ชันหน้าอัปโหลดเอกสาร ฟังก์ชันสถานะเอกสาร ฟังก์ชันอัปเดตเอกสารและฟังก์ชันดูรายละเอียดเอกสาร

### 4.2.1 โครงสร้างฐานข้อมูล



รูปที่ 32 ฐานข้อมูลของเว็บแอปพลิเคชันระบบติดตามเอกสาร

#### 4.2.1.1 ฐานข้อมูลหน้าลงทะเบียนและหน้าล็อกอิน

ฐานข้อมูลฟังก์ชันลงทะเบียนและล็อกอินจะประกอบด้วย id, fname, lname, email, passwords, roleuser ซึ่งฟังก์ชันของล็อกอินนั้นจะใช้แค่ email, passwords ในการกรอกข้อมูล

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
fname	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
lname	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
passwords	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
email	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
roleuser	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

รูปที่ 33 ฐานข้อมูลของหน้าลงทะเบียนและหน้าล็อกอิน

10	Jarunay	Thanatavee	student	6210110459@email.psu.ac.th	student
15	Ja	Thanatavee	\$2b\$10\$VxB9XoTB9995N8CfZ08gf.9gJzuSbQyp...	6210110800@email.psu.ac.th	student

รูปที่ 34 ตัวอย่างการเก็บข้อมูลของหน้าลงทะเบียนและหน้าล็อกอิน

#### 4.2.1.2 ฐานข้อมูลเก็บไฟล์เอกสาร

ฐานข้อมูลเก็บไฟล์เอกสารจะประกอบด้วย id, topic, username, userfile, date, decs, status\_id, usersy\_id ซึ่ง status\_id จะเป็นตัวแปรที่ใช้แสดงสถานะเอกสาร, usersy\_id จะเป็นตัวแปรที่ใช้สำหรับดูชื่อผู้ส่งไฟล์เอกสาร

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
topic	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
username	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
userfile	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
date	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
decs	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
status_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
usersy_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

รูปที่ 35 ฐานข้อมูลของหน้าหลัก

#### 4.2.1.3 ฐานข้อมูลสถานะเอกสาร

ฐานข้อมูลสถานะเอกสารจะประกอบด้วย id, statusfile ซึ่งจะประกอบสถานะเอกสาร 2 สถานะคือสถานะรอการดำเนินและสถานะเอกสารสำเร็จ

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
status_file	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

รูปที่ 36 ฐานข้อมูลของสถานะเอกสาร

## 4.2.2 การออกแบบโดยใช้ REST API

โครงการนี้มีการพัฒนา REST API เพื่อที่จะใช้ในการสื่อสารแลกเปลี่ยนข้อมูลผ่าน nodeJS โดยการทำงานของ REST API จะอาศัย URL ของการร้องขอเพื่อค้นหาและประมวล แล้วกลับข้อมูลในรูปแบบ JSON ซึ่งการพัฒนาขึ้นฟังก์ชันการใช้งาน 7 ฟังก์ชัน ได้แก่ ฟังก์ชันลงทะเบียน ฟังก์ชันล็อกอิน ฟังก์ชันหน้าหลัก ฟังก์ชันหน้าอัปโหลดเอกสาร ฟังก์ชันสถานะเอกสาร ฟังก์ชันอัปเดตเอกสารและฟังก์ชันดูรายละเอียดเอกสาร

```
1 const express = require('express');
2 const router = new express.Router();
3 const conn = require('../db/conn');
4 const multer = require('multer');
5 const moment = require('moment');
6
7 const bodyParser = require('body-parser');
8 var jsonParser = bodyParser.json();
9 const bcrypt = require('bcrypt');
10 const saltRounds = 10;
11 var jwt = require('jsonwebtoken');
12 const secret = 'DocumentSystem';
13 // const session = require('express-session');
14
15
16 // file storage config
17 > var fileconfig = multer.diskStorage({
25 });
26
27
28 //file filter
29 > const isFile = (req, file, callback) => {
35 }
36
37 > var upload = multer({
40 })
41
42
43 // register userdata
44 > router.post('/register', upload.single('photo'), (req, res) => {
73 })
74
75 // get user data
76 > router.get('/getdata', (req, res) => {
90 })
91
92 // Delete user
93 > router.delete('/delete/:id', (req, res) => {
107 })
108
109 //get single user
110 > router.get('/induser/:id', (req, res) => {
111   const { id } = req.params;
112   try {
113     conn.query('SELECT * FROM usersdata WHERE id=?', [id], (err, result) => {
```

รูปที่ 37 ตัวอย่างโค้ดที่ใช้สร้างเชื่อมต่อฐานข้อมูล เพื่อจะทำการส่งและรับ API

### 4.2.2.1 REST API ของฟังก์ชันลงทะเบียน

REST API ของฟังก์ชันลงทะเบียน จะมีการเลือกใช้คำสั่ง 'INSERT INTO usersy' เนื่องจากการเพิ่มข้อมูลใหม่เพื่อนำไปใช้ในการล็อกอินเข้าสู่ระบบ โดยเราจะมีการนำ hash ใช้ในการแปลง password เพื่อให้ระบบมีความปลอดภัยมากขึ้น

```
165 //sign in
166 router.post('/registeruser', jsonParser, function (req, res, next) {
167   // const { email, ffname, llnname, passwords, roleuser } = req.body;
168
169   bcrypt.hash(req.body.passwords, saltRounds, function (err, hash) {
170     conn.query('INSERT INTO usersy (ffname, llnname, email, passwords, roleuser) VALUES (?, ?, ?, ?, ?)',
171       [req.body.ffname, req.body.llname, req.body.email, hash, req.body.roleuser],
172       function (err, result, fields) {
173         if (err) {
174           res.json({ status: 'error', message: err })
175           return
176         }
177         res.json({ status: 'OK' })
178       })
179     });
180   });
```

รูปที่ 38 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันลงทะเบียน

#### 4.2.2.2 REST API ของฟังก์ชันการล็อกอิน

REST API ของฟังก์ชันล็อกอิน จะมีการเลือกใช้คำสั่ง 'SELECT \* FROM usersy WHERE email=?' เพื่อในการตรวจสอบว่า email กับ password ที่กรอกมาตรงกับฐานข้อมูลที่เก็บไว้หรือไม่ ถ้าตรงเราจะให้ token เพื่อเป็นยืนยันตัวตนและเข้าสู่หน้าหลักได้

```
router.post('/loginuser', jsonParser, function (req, res, next) {
  // const { email, ffname, lname, passwords, roleuser } = req.body;

  conn.query('SELECT * FROM usersy WHERE email=?', [req.body.email], function (err, users, fields) {
    if (err) {
      res.json({ status: 'error', message: err });
      return;
    }
    if (users.length == 0) {
      res.json({ status: 'error', message: 'no user found' });
      return;
    }
    bcrypt.compare(req.body.passwords, users[0].passwords, function (err, isLogin) {
      if (isLogin) {
        var token = jwt.sign({ email: users[0].email }, secret, { expiresIn: '1h' });
        res.json({ status: 'ok', message: 'login success', token });
      } else {
        res.json({ status: 'error', message: 'login failed' });
      }
    });
  });
  // res.json({ status: 'OK' })
})

router.post('/authuser', jsonParser, function (req, res, next) {
  try {
    const token = req.headers.authorization.split(' ')[1]
  }
})
```

รูปที่ 39 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันล็อกอิน

#### 4.2.2.3 REST API ของฟังก์ชันการอัปโหลดไฟล์

REST API ของฟังก์ชันล็อกอิน จะมีการเลือกใช้คำสั่ง 'INSERT INTO usersdata' เนื่องจากการเพิ่มข้อมูลใหม่ในการเก็บข้อมูลของไฟล์เอกสาร โดยจะมีตัวแปร filename เป็นตัวแปรที่ใช้เก็บไฟล์เอกสารซึ่งตัวแปรนี้จะมาจากการเรียกใช้ upload.single ซึ่งจะเป็นตัวกำหนดฟิลด์เตอร์ที่เก็บไฟล์และชื่อไฟล์ที่ใช้ในการเก็บในฐานข้อมูล

```
// register userdata
router.post('/register', upload.single('photo'), (req, res) => {
  // console.log(req.body);
  const { tname, fname, dname } = req.body;
  const { filename } = req.file;
  const { status } = req.body;

  if (!fname || !filename || !tname || !dname || !status) {
    res.status(422).json({ status: 422, message: "fill all the details" })
  }

  try {
    let date = moment(new Date()).format('YYYY-MM-DD hh:mm:ss');

    conn.query('INSERT INTO usersdata SET ?', { topic: tname, username: fname, userfile: filename, decs: dname, status_id: status, date: date }, (err, result) => {
      if (err) {
        console.log("error")
      } else {
        console.log("data added")
        res.status(201).json({ status: 201, data: req.body })
      }
    })
  }
})
```

รูปที่ 40 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันอัปโหลดไฟล์

```

// file storage config
var fileconfig = multer.diskStorage({
  destination: (req, file, callback) => {
    //Store uploaded files
    callback(null, "./uploads");
  },
  filename: (req, file, callback) => {
    callback(null, `file-${Date.now()}.${file.originalname}`)
  }
});

//file filter
const isFile = (req, file, callback) => {
  if (file.mimetype.startsWith("image")) {
    callback(null, true)
  } else {
    callback(null, Error("only file is allowed"))
  }
}

var upload = multer({
  storage: fileconfig,
  fileFilter: isFile
});

```

รูปที่ 41 ตัวอย่างโค้ดของ upload.single

#### 4.2.2.4 REST API ของฟังก์ชันหน้าหลัก

REST API ของฟังก์ชันหน้าหลัก จะมีการเลือกใช้คำสั่ง ‘SELECT ... FROM ... INNER JOIN ...’ เนื่องจากฐานข้อมูลของการเก็บไฟล์เอกสารนั้นมีคีย์จากตารางอื่นมามีความสัมพันธ์ด้วย จึงทำให้การเรียกใช้นั้นต้องมีการเรียกใช้แบบรวบรวมหลายตาราง เพื่อให้แสดงผลตามที่หน้าหลักต้องแสดง และสำหรับหน้าหลักของนักศึกษาจะมีการเรียกใช้ REST API ลบข้อมูลเนื่องจาก API จะมีการลบข้อมูลตาม id ของฐานข้อมูลเก็บไฟล์เอกสาร

```

// register userdata
> router.post("/register", upload.single("photo"), (req, res) => {
  // get user data
  router.get("/getdata", (req, res) => {
    try {
      conn.query("SELECT usersdata.id, usersdata.topic, usersdata.username, usersdata.date, usersdata.decs,statusfile.status_file F
      (err, result) => {
        if (err) {
          console.log("error")
        } else {
          console.log("data get")
          res.status(201).json({ status: 201, data: result })
        }
      })
    } catch (error) {
      res.status(422).json({ status: 422, error })
    }
  })
});

```

รูปที่ 42 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันหน้าหลัก

```
// delete user
router.delete("/delete/:id", (req, res) => {
  const { id } = req.params;
  try {
    conn.query("DELETE FROM usersdata WHERE id = '${id}'", (err, result) => {
      if (err) {
        console.log("error")
      } else {
        console.log("data delete")
        res.status(201).json({ status: 201, data: result })
      }
    })
  } catch (error) {
    res.status(422).json({ status: 422, error })
  }
})
```

รูปที่ 43 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันการลบข้อมูลเอกสาร

#### 4.2.2.5 REST API ของฟังก์ชันดูสถานะเอกสาร

REST API ของฟังก์ชันดูสถานะเอกสาร จะมีการเลือกใช้คำสั่ง 'SELECT ... FROM ... INNER JOIN ...' เนื่องจากฐานข้อมูลของเก็บไฟล์เอกสารมีคีย์ที่เชื่อมสัมพันธ์กับตารางสถานะเอกสาร จึงทำให้สามารถเรียกดูข้อมูลได้ผ่านทางฐานข้อมูลของเก็บไฟล์เอกสารและตัวแปร statusfile.status\_file จะเป็นการเรียกใช้ข้อมูลแสดงผลตามที่เก็บข้อมูลไว้ในตารางสถานะ

```
// get user data
router.get("/getdata", (req, res) => {
  try {
    conn.query("SELECT usersdata.id, usersdata.topic, usersdata.username, usersdata.date, usersdata.decs, statusfile.status_file FROM usersdata INNER JOIN statusfile ON usersdata.id = statusfile.user_id", (err, result) => {
      if (err) {
        console.log("error")
      } else {
        console.log("data get")
        res.status(201).json({ status: 201, data: result })
      }
    })
  } catch (error) {
    res.status(422).json({ status: 422, error })
  }
})
```

รูปที่ 44 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันดูสถานะเอกสาร

#### 4.2.2.6 REST API ของฟังก์ชันอัปเดตเอกสาร

REST API ของฟังก์ชันอัปเดตเอกสาร จะมีการเลือกใช้คำสั่ง 'UPDATE' โดยตัวแปรที่มีการอัปเดตนั้นคือ ตัวแปร filename เป็นตัวแปรที่ใช้สำหรับอัปเดตเอกสารที่เจ้าหน้าที่อัปเดตมาให้และตัวแปร status เป็นตัวแปรตามเมื่อมีการอัปเดตไฟล์เอกสารแล้วสถานะของตัวแปรนี้จะเปลี่ยนไปด้วย ซึ่งการอัปเดตจะมีตัวแปร id เป็นตัวกำหนดข้อมูล

```

router.put("/update/:id", upload.single("photo"), (req, res) => {
  const { id } = req.params;
  const { filename } = req.file;
  const { status } = req.body;
  (alias) const conn: mysql.Connection
  import conn
  try {
    conn.query("UPDATE usersdata SET userfile = ?, status_id = ? WHERE id = ?", [filename, status, id], (err, result) => {
      if (err) {
        console.log("error")
      } else {
        console.log("update data")
        res.status(201).json({ status: 201, data: result })
      }
    })
  } catch (error) {
    res.status(422).json({ status: 422, error })
  }
})

```

รูปที่ 45 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันอัปเดตเอกสาร

#### 4.2.2.7 REST API ของฟังก์ชันดูรายละเอียดของเอกสาร

REST API ของฟังก์ชันดูรายละเอียดเอกสาร ฟังก์ชันนี้จะมีการเรียกใช้ API แบบ id เป็นตัวแปรที่ใช้กำหนดการดึงข้อมูลนั้นมาแสดง

```

//get single user
router.get("/induser/:id", (req, res) => {
  const { id } = req.params;
  try {
    conn.query("SELECT * FROM usersdata WHERE id=?", [id], (err, result) => {
      if (err) {
        console.log("error")
      } else {
        console.log("data show")
        res.status(201).json({ status: 201, data: result })
      }
    })
  } catch (error) {
    res.status(422).json({ status: 422, error })
  }
})

```

รูปที่ 46 ตัวอย่างโค้ดที่ใช้ API ฟังก์ชันดูรายละเอียดเอกสาร



## บทที่ 5

### สรุปและข้อเสนอแนะ

#### 5.1 สรุป

เว็บแอปพลิเคชันระบบติดตามเอกสารจะเข้ามาช่วยเหลือนักศึกษา อาจารย์และเจ้าหน้าที่ในองค์กรให้สามารถติดตามเอกสารได้ โดยมีฟังก์ชันการทำงานดังนี้ ฟังก์ชันการอัปโหลดเอกสาร จะเข้ามาช่วยในการเก็บรวบรวมไฟล์ได้เป็นระบบมากขึ้นและสามารถค้นหาเอกสารต่างๆ ได้โดยไม่ใช้เวลานาน ฟังก์ชันดูสถานะเอกสารและฟังก์ชันการอัปเดตเอกสาร จะเข้ามาช่วยในการติดตามเอกสารว่ามีขั้นตอนถึงไหนแล้วเพื่อให้สะดวกต่อนักศึกษาไม่ต้องมาสอบถามเองกับเจ้าหน้าที่ และเจ้าหน้าที่ไม่ต้องจัดส่งเอกสารให้นักศึกษาในรูปแบบของขนส่งเอกสารด้วย เนื่องจากอาจจะเกิดข้อผิดพลาดทางการขนส่งหรือการส่งเอกสารผิดเป็นได้ และฟังก์ชันดูรายละเอียดเอกสารจะทำให้เพื่อความสะดวกในส่วนของอาจารย์และเจ้าหน้าที่เพื่อดูว่าเอกสารของนักศึกษาคนนี้เรียบร้อยหรือไม่ หรือส่งเอกสารถูกต้องหรือไม่ ซึ่งการทำงานของระบบโดยภาพรวมทั้งหมดสามารถทำงานได้ตรงตามออกแบบไว้ยกเว้นการล็อกอินผู้ใช้งานตอนนี้มีความติดขัดและยังไม่เสถียรตามต้องการ

#### 5.2 ข้อเสนอแนะ

1. ปัจจุบันระบบยังไม่สามารถแบ่งผู้ใช้งานในการเข้าระบบได้ จึงไม่สามารถรู้ถึงความเสถียรของการทำงานของแต่ละผู้ใช้
2. การล็อกอินเข้าสู่ระบบยังมีความไม่เสถียรในการเรียกใช้ REST API ถึงทำให้ไม่สามารถดึงชื่อของผู้ใช้งานในระบบได้

## บรรณานุกรม

- [1] “วิทยาลัยวิทยาศาสตร์ดิจิทัล”, cds.psu.ac.th, [Online]. Available: <https://cds.psu.ac.th/index.php/about-us/about>. [Accessed 2022, Jan 18].
- [2] “PSU.College.of.Digital.Science”, facebook.com, [Online]. Available: <https://www.facebook.com/groups/PSU.College.of.Digital.Science>. [Accessed 2022, Jan 18].
- [3] “JavaScript และ ES6 คืออะไร”, devjourneys.com, [Online]. Available: <https://devjourneys.com/2020/04/02/javascript-และ-es6-คืออะไร>. [Accessed 2022, Jan 16]
- [4] Sirawit. (2022, Jan 18). JavaScript สมัยใหม่ (Modern JavaScript) ในปี 2019, [Online]. Available: <https://medium.com/@sirawit/javascript>.
- [5] dev-devportal.engpsu.ac.th, [Online]. Available: <https://dev-devportal.eng.psu.ac.th>.
- [6] “Database คืออะไร”, mindphp.com, [Online]. Available: <https://www.mindphp.com> [Accessed 2022, Jan 19]
- [7] “RDBMS (ระบบจัดการฐานข้อมูล)”, searchdatamanagement.techtarget.com, [Online]. Available: <https://www.techtarget.com/searchdatamanagement/definition/RDBMS-relational-database-management-system>. [Accessed 2022, Jan 19]
- [8] “หลักการของ ACID”, bnalive.wordpress.com, [Online]. Available: <https://bnalive.wordpress.com> [Accessed 2022, Jan 19]
- [9] “React js คือ”, code-fin-nua.com, [Online]. Available: <https://code-fin-nua.com/blog/react>. [Accessed 2023, Jan 29]
- [10] “Node js คือ”, glurgeek.com, [Online]. Available: <https://www.glurgeek.com/education/node-js>. [Accessed 2023, Jan 29]
- [11] “Node js คือ”, marcuscode.com, [Online]. Available: <http://marcuscode.com/tutorials/nodejs/introducing-nodejs>. [Accessed 2023, Jan 29]

- [12]“mysql workbench คือ”, itgenius.co.th, [Online]. Available: [https://www.itgenius.co.th/article/\(MySQL\).html](https://www.itgenius.co.th/article/(MySQL).html) [Accessed 2023,Jan 29]
- [13]“JWT คือ ”, medium.com, [Online]. Available: <https://medium.com/rootusercc/json-web-token>. [Accessed 2023,Mar 3]

## ภาคผนวก

