



รายงานโครงการวิทยาการคอมพิวเตอร์

Disboard Website

ผู้จัดทำ

นายอภิษฐ์ เดชสุวรรณ

รหัสประจำตัวนิสิต 6310406370

เสนอ

ผศ. ดร. ผกาเกษ วัตูยา

ผศ. ดร. ชาศริต วัชรโรภาส

ผศ. ดร. ธรรมกร แซ่ตั้ง

รายวิชา 01418499 โครงการวิทยาการคอมพิวเตอร์

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

ภาคปลาย ปีการศึกษา 2566

โครงการวิทยาการคอมพิวเตอร์

ชื่อเรื่อง

Disboard Website

ผู้จัดทำ

นายอภิษฐ์ เดชสุวรรณ รหัสประจำตัวนิสิต 6310406370

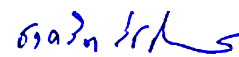
อาจารย์ที่ปรึกษาหลัก



ผศ. ดร. ผกาเกษ วัฒยา

07/05/67

อาจารย์ที่ปรึกษาร่วม



ผศ. ดร. ชاکริต วิชโรภาส

07/05/67

อาจารย์ที่ปรึกษาร่วม



ผศ. ดร. ธรรมกร แซ่ตั้ง

08/05/67

หัวหน้าภาควิชาวิทยาการคอมพิวเตอร์



อ. สมโชค เรืองอิทธินันท์

28/6/67

นายอภิษฐ์ เดชสุวรรณ. 2567. **Disboard Website**. โครงการวิทยาการคอมพิวเตอร์
วิทยาศาสตร์บัณฑิต ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.

อาจารย์ที่ปรึกษา: ผศ.ดร.ผกาเกษ วัตุยา

บทคัดย่อ

โครงการนี้จัดทำขึ้นโดยมีวัตถุประสงค์ให้อาจารย์สามารถติดต่อสื่อสารกับแพทย์เพื่อให้ความช่วยเหลือในการดำเนินงานวิจัย ได้อย่างมีประสิทธิภาพ และค้นหาผู้ที่มีความสนใจสำหรับการดำเนินงานวิจัยร่วมกัน ด้วยการสร้างเว็บไซต์ด้วยภาษาโปรแกรม PHP รวมทั้งการวิเคราะห์และออกแบบระบบซึ่งประกอบด้วยคุณสมบัติ ได้แก่ การสร้างบัญชีผู้ใช้ การเข้าสู่ระบบ การสร้างโพสต์ เข้าถึงโพสต์ การแสดงความคิดเห็นบนโพสต์ การแก้ไขโพสต์ของตนเอง การแสดงข้อมูลในโปรไฟล์ของผู้ใช้แต่ละคน การแก้ไขข้อมูลในโปรไฟล์ของตนเอง โดยกำหนดให้แพทย์สามารถโพสต์ปัญหาและแสดงความคิดเห็นบนโพสต์ของตนเองได้เท่านั้น ในขณะที่อาจารย์สามารถโพสต์ประกาศ รวมถึงแสดงความคิดเห็นบนโพสต์ของผู้ใช้ทุกคน จึงสรุปได้ว่าโครงการนี้ช่วยอำนวยความสะดวกให้กับแพทย์และอาจารย์ในการติดต่อสื่อสาร ลดอุปสรรคระหว่างการดำเนินงานวิจัย และเพิ่มโอกาสในการดำเนินงานวิจัยร่วมกัน

Proposal

This project was created with the aim of allowing professors to communicate with doctors to help in working on research. effectively and find people who are interested in conducting joint research. By creating a website with the PHP programming language, including system analysis and design, which includes features such as creating user accounts. login, creating a post, viewing a post, commenting on a post, editing your own posts displaying each user's profile, and editing self-information. By allowing doctors to only post problems and comments on their own posts. While teachers can also post announcements and comments on every user's post. Therefore, it can be concluded that this project could help with communication between doctors and professors. Reduce troubles during research operations and increase opportunities to work on research together.

กิตติกรรมประกาศ

โครงการวิทยาการคอมพิวเตอร์นี้ ประสบความสำเร็จได้จากการได้รับคำแนะนำ ความอนุเคราะห์ และการสนับสนุนจากอาจารย์ที่ปรึกษาโครงการ ผศ. ดร. ผกาเกษ วัฒยา ที่ได้ให้ความช่วยเหลือในการดำเนินงาน รวมถึง ผศ. ดร. ชาศริต วัชรโรภาส และ ผศ. ดร. ธรรมกร แซ่ตั้ง อาจารย์ที่ปรึกษาโครงการร่วม ซึ่งได้ให้ความช่วยเหลือในการวิเคราะห์และออกแบบระบบ รวมถึงให้ข้อคิดและคำแนะนำแก่ผู้จัดทำโครงการตลอดระยะเวลาการดำเนินงาน และผู้ที่ได้ให้ความช่วยเหลือในการดำเนินงานทุกท่าน ผู้จัดทำขอขอบคุณเป็นอย่างยิ่ง

ผู้จัดทำคาดหวังว่าโครงการและรายงานนี้จะเป็นประโยชน์ต่อผู้ที่สนใจในการออกแบบและพัฒนาเว็บไซต์ การวิเคราะห์และออกแบบระบบ และการจัดทำเอกสารประกอบโครงงาน หากรายงานฉบับนี้มีข้อผิดพลาดประการใด ผู้จัดทำขออภัยมา ณ ที่นี้

นายอภิษฐ์ เดชสุวรรณ

ผู้จัดทำ

1 พฤษภาคม 2567

สารบัญ

บทคัดย่อ.....	3
Proposal	3
กิตติกรรมประกาศ	4
สารบัญ	5
บทที่ 1 บทนำ.....	9
1.1 ที่มาและความสำคัญ	9
1.2 วัตถุประสงค์.....	9
1.3 ขอบเขตการดำเนินงาน.....	9
1.4 ประโยชน์ที่คาดว่าจะได้รับ	10
1.5 เครื่องมือในการพัฒนาระบบ	10
1.6 แผนการดำเนินงาน	11
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	12
2.1 Bootstrap.....	12
2.2 Docker	12
2.3 jQuery.....	13
2.4 PHP Data Objects.....	13
2.5 SESSION	14
บทที่ 3 การวิเคราะห์และออกแบบระบบ.....	15
3.1 การวิเคราะห์ระบบ.....	15
3.2 กระบวนการดำเนินงานของระบบ	16

3.3 คุณสมบัติของระบบ	17
3.4 ขั้นตอนการดำเนินงานของแต่ละคุณสมบัติของระบบ.....	18
3.4.1 การสร้างบัญชีผู้ใช้	18
3.4.2 การเข้าสู่ระบบ	19
3.4.3 การสร้างโพสต์	20
3.4.4 การเข้าถึงโพสต์.....	21
3.4.5 การแสดงความคิดเห็นบนโพสต์ปัญหาทางวิจัย	22
3.4.6 การแก้ไขโพสต์ของผู้ใช้.....	23
3.4.7 การเข้าถึงข้อมูลในโปรไฟล์ของผู้ใช้.....	24
3.4.8 การแก้ไขข้อมูลในโปรไฟล์ของผู้ใช้.....	25
3.5 การออกแบบฐานข้อมูล.....	26
3.5.1 แผนภาพฐานข้อมูล	26
3.5.2 ความสัมพันธ์ระหว่างตารางของฐานข้อมูล.....	27
บทที่ 4 การพัฒนาระบบ.....	31
4.1 การเข้าและออกจากระบบ	31
4.1.1 การสร้างบัญชีผู้ใช้	31
4.1.2 การเข้าสู่ระบบ	32
4.1.3 การออกจากระบบ	32
4.2 การโพสต์	32
4.2.1 การสร้างโพสต์	33
4.2.2 การแก้ไขโพสต์.....	33
4.2.3 การเข้าถึงโพสต์.....	34
4.3 การแสดงความคิดเห็น.....	35

4.4 การเข้าถึงและแก้ไขโปรไฟล์.....	36
4.4.1 การเข้าถึงโปรไฟล์	36
4.4.2 การแก้ไขโปรไฟล์	37
บทที่ 5 การทดสอบระบบ	38
5.1 การเชื่อมต่อฐานข้อมูล.....	38
5.2 การสร้างตารางฐานข้อมูล.....	39
5.2.1 ตาราง users.....	39
5.2.2 ตาราง posts	40
5.2.3 ตาราง files.....	40
5.2.4 ตาราง comments.....	41
5.3 การเริ่มต้นระบบ.....	41
5.3.1 Web Server	41
5.3.2 Database Server.....	41
บทที่ 6 บทสรุป ปัญหา ข้อเสนอแนะ และแนวทางการพัฒนาต่อยอด	42
6.1 บทสรุป	42
6.2 ปัญหา	42
6.3 ข้อเสนอแนะ	42
6.4 แนวทางการพัฒนาต่อยอด	42
บรรณานุกรม.....	43
ภาคผนวก ก การติดตั้งระบบ	44
ภาคผนวก ข วิธีการใช้งานระบบ.....	45
1. การสร้างบัญชีผู้ใช้.....	45
2. การเข้าสู่ระบบ.....	47

3. การโพสต์.....	48
4. การแสดงความคิดเห็น.....	52
5. การเข้าถึงโปรไฟล์.....	55
6. การแก้ไขข้อมูลยืนยันตัวตนและการออกจากระบบ	58

บทที่ 1

บทนำ

บทนำนี้ประกอบด้วยที่มาและความสำคัญของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตการดำเนินงานโดยระบุลักษณะของโครงการอย่างสังเขป ประโยชน์ที่คาดว่าจะได้รับจากโครงการ เครื่องมือในการพัฒนาโครงการ และแผนการดำเนินงานโดยระบุลำดับการดำเนินงานและระยะเวลาที่ใช้ในแต่ละขั้นตอน

1.1 ที่มาและความสำคัญ

ในปัจจุบันการติดต่อสื่อสารระหว่างบุคคล ณ ต่างสถานที่ มีความสะดวกและรวดเร็ว อันเนื่องมาจากระบบอินเทอร์เน็ต โดยผู้ใช้งานสามารถติดต่อสื่อสารกับผู้อื่นผ่านการใช้สื่อสังคม ซึ่งในปัจจุบันได้รับความนิยมเป็นอย่างสูง สื่อสังคมจึงถูกนำมาใช้ในการติดต่อสื่อสารในทุกสถานการณ์ รวมถึงการสื่อสารระหว่างการดำเนินงานวิจัยของแพทย์นักวิจัย และอาจารย์มหาวิทยาลัย ซึ่งต้องการความถูกต้องและเหมาะสม แต่สื่อสังคมที่มีข้อมูลการสื่อสารกับบุคคลอื่น เช่น ครอบครัว มิตร หรือข่าวสาร ทำให้การสนทนาสำหรับการดำเนินงานลดประสิทธิภาพลง

การมีวิธีการติดต่อสื่อสารเพื่อการดำเนินงานวิจัยระหว่างแพทย์นักวิจัยและอาจารย์มหาวิทยาลัย โดยเฉพาะ เพื่อให้อาจารย์สามารถให้ความช่วยเหลือ หรือให้ความคิดเห็นแก่แพทย์นักวิจัยได้อย่างมีประสิทธิภาพ จึงมีความสำคัญเป็นอย่างยิ่ง ด้วยเหตุผลดังกล่าว โครงการ Disboard จึงมีลักษณะรูปแบบเป็นพื้นที่ในสนทนาที่มีความเป็นส่วนตัวระหว่างแพทย์และอาจารย์ โดยชื่อของโครงการได้แปรผันมาจากคำศัพท์ discussion board หมายถึง กระดานสนทนา เพื่อเพิ่มประสิทธิภาพในดำเนินงานวิจัยให้กับแพทย์ต่อไป

1.2 วัตถุประสงค์

1. เพื่อให้อาจารย์สามารถติดต่อสื่อสารกับแพทย์เพื่อให้ความช่วยเหลือในการดำเนินงานวิจัย
2. เพื่อให้แพทย์สามารถดำเนินงานวิจัยได้อย่างมีประสิทธิภาพ
3. เพื่อค้นหาผู้ที่มีความสนใจสำหรับการดำเนินงานวิจัยร่วมกันระหว่างแพทย์และอาจารย์

1.3 ขอบเขตการดำเนินงาน

โครงการ Disboard ดำเนินงานภายใต้รายวิชา 01418499 โครงการวิทยาการคอมพิวเตอร์ โดยมีระยะเวลาการดำเนินงานระหว่างเดือนธันวาคม พุทธศักราช 2566 ถึงเดือนพฤษภาคม พุทธศักราช 2567 เพื่อให้คณาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ สามารถให้ความช่วยเหลือในการดำเนินงานวิจัยของแพทย์ผู้วิจัย ณ วิทยาลัยแพทยศาสตร์พระมงกุฎเกล้า โดยมีลักษณะของโครงการอย่างสังเขปดังนี้

1. การโพสต์ปัญหางานวิจัย

แพทย์สามารถโพสต์ปัญหางานวิจัยเพื่อขอความช่วยเหลือจากอาจารย์

2. การแสดงความคิดเห็น

อาจารย์สามารถแสดงความคิดเห็นในโพสต์ปัญหางานวิจัย หรือให้ความช่วยเหลือ

3. การเข้าถึงข้อมูลทางวิชาชีพของผู้ใช้งาน

ผู้ใช้งานสามารถเรียกดูข้อมูลทางวิชาชีพของผู้ใช้งานท่านอื่น เพื่อโอกาสการดำเนินงานร่วมกัน

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. อาจารย์สามารถให้ความช่วยเหลือแก่แพทย์ได้มากขึ้น
2. แพทย์สามารถดำเนินงานวิจัยได้อย่างมีประสิทธิภาพมากขึ้น
3. แพทย์และอาจารย์สามารถทราบถึงประวัติและความสนใจของแต่ละบุคคล

1.5 เครื่องมือในการพัฒนาระบบ

1. เครื่องมือในการวิเคราะห์และออกแบบระบบ

เว็บแอปพลิเคชัน diagrams.net สำหรับการสร้างแผนภาพขั้นตอนการวิเคราะห์และออกแบบระบบ

2. เครื่องมือในการพัฒนาระบบ

- โปรแกรม Visual Studio Code สำหรับการเขียนโปรแกรมในการพัฒนาระบบ

- โปรแกรม Xampp สำหรับการจำลองเว็บไซต์เพื่อการแสดงผลระหว่างการพัฒนา

3. เครื่องมือในการทดสอบระบบ

โปรแกรม Docker สำหรับการทดสอบและติดตั้งเว็บไซต์เพื่อทดสอบการใช้งาน

1.6 แผนการดำเนินงาน

โครงการ Disboard มีการลำดับการดำเนินงานดังต่อไปนี้

1. การวางแผนการดำเนินงาน
2. ค้นหาข้อมูลที่เกี่ยวข้อง
3. การวิเคราะห์และออกแบบระบบ
4. การพัฒนาระบบ
5. การทดสอบระบบ
6. การสรุปผลและจัดทำเอกสารประกอบโครงการ

โดยแสดงแผนขั้นตอนการดำเนินงานและระยะเวลาที่ใช้ดังตาราง 1.1

#	ระยะเวลาการดำเนินงาน พุทธศักราช 2566 - 2567					
	ธันวาคม	มกราคม	กุมภาพันธ์	มีนาคม	เมษายน	พฤษภาคม
1	-----	-----				
2	-----	-----				
3		-----				
4		-----	-----	-----	-----	
5				-----	-----	
6				-----	-----	---

ตาราง 1.1 แผนการดำเนินงานโครงการ

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องดังต่อไปนี้จะได้กล่าวถึงข้อมูลเบื้องต้น และองค์ประกอบที่ได้นำมาใช้ในการพัฒนาโครงการ โดยเรียงลำดับชื่อศัพท์และทฤษฎีตามลำดับอักษร

2.1 Bootstrap

Bootstrap คือเครื่องมือชุดคำสั่งสำหรับการพัฒนาส่วนแสดงผลของเว็บไซต์ ซึ่งประกอบไปด้วยชุดคำสั่งภาษา HTML CSS และ JavaScript โดยมีจุดมุ่งหมายเพื่อลดความซับซ้อน และเพิ่มความสวยงามในการพัฒนาส่วนแสดงผลของเว็บไซต์ โดยผู้ใช้งานสามารถกำหนดสี ขนาดและรูปแบบตัวอักษร การจัดเรียงข้อมูลในแสดงผล โดยส่วนใหญ่ผู้ใช้งานจะทำการเขียนในภาษา HTML เพื่อเรียกคลาสในภาษา CSS ซึ่งจากคลังเอกสารของ Bootstrap

Bootstrap มีความสำคัญในการออกแบบและพัฒนาเว็บไซต์ซึ่งการแสดงผลปรับเปลี่ยนไปตามขนาดของจอภาพผู้ใช้ โดยผู้ใช้งานสามารถเรียกการใช้งาน Breakpoints เพื่อกำหนดขนาดของจอภาพซึ่งแบ่งขอบเขตดังต่อไปนี้

โครงการนี้ได้นำ Bootstrap เวอร์ชัน 5.3.3 มาใช้ในการพัฒนาส่วนแสดงผลของเว็บไซต์ ซึ่งประกอบด้วยภาษา HTML ในเว็บไซต์แต่ละหน้า ภาษา CSS ในสำหรับ component ที่ได้สร้างขึ้น และชุดคำสั่งภาษา JavaScript สำหรับการเปลี่ยนแปลงตามธีมสีโทนของเว็บไซต์ระหว่างโทนสว่างและโทนมืด ตามที่ผู้ใช้งานได้ตั้งค่าในอุปกรณ์ของตนเอง

2.2 Docker

Docker คือแพลตฟอร์มซอฟต์แวร์สำหรับการสร้าง ทดสอบ และติดตั้งแอปพลิเคชัน โดย Docker จะบรรจุซอฟต์แวร์ลงในหน่วยที่เป็นมาตรฐาน เรียกว่า container เพื่อให้ผู้ใช้งานสามารถติดตั้งและใช้งานแอปพลิเคชันได้ในสภาพแวดล้อมที่แตกต่างกันของแต่ละระบบ โดยแต่ละ container มีการทำงานเป็นอิสระจากกันและกัน มีเพียงการใช้ kernel ของระบบปฏิบัติการร่วมกันเท่านั้น คล้ายกับซอฟต์แวร์ virtual machine

องค์ประกอบพื้นฐานของ Docker ได้แก่ docker file ซึ่งเป็นเอกสารคำสั่งในการสร้าง Docker image หรือแม่แบบในการสร้าง container เพื่อให้แอปพลิเคชันเรียกใช้งาน และจำลองสภาพแวดล้อมของแอปพลิเคชันนั้นด้วย และ Docker container ซึ่งบรรจุแอปพลิเคชันและสภาพแวดล้อมที่จำเป็นการทำงาน

Docker ได้ถูกมาใช้ในการพัฒนาและทดสอบระบบของโครงการ โดยเขียนคำสั่งเพื่อติดตั้งองค์ประกอบที่จำเป็นในการพัฒนาระบบ เช่น ชุดคำสั่งสำหรับการดำเนินการกับฐานข้อมูล และการจำลองสภาพแวดล้อมสำหรับการติดต่อไปยังระบบ ได้แก่ ชื่อผู้ใช้ รหัสผ่าน ฐานข้อมูล และโดเมนของเว็บไซต์นั้น

2.3 jQuery

jQuery คือไลบรารีในภาษา JavaScript ในการจัดการกับองค์ประกอบของการแสดงผลบนภาษา HTML ในการเปลี่ยนเส้นทาง และการเปลี่ยนแปลงข้อมูลโดยไม่มีเปลี่ยนแปลงหน้าเว็บไซต์ โดยการสร้างสถานะของส่วนแสดงผลนั้นในสถานะ ready เพื่อรับพารามิเตอร์ในการดำเนินงานกับส่วนแสดงผลนั้น

การดำเนินงานของ jQuery สามารถทำได้ด้วยการ click หรือการกดเลือก การ callback เรียกฟังก์ชันภายในฟังก์ชันดังกล่าว การใช้ each เลือกข้อมูลแต่ละข้อมูล และการ change เปลี่ยนแปลงข้อมูลหรือการทำงานของส่วนแสดงนั้น

jQuery ได้นำมาใช้ในการแสดงผลภาพโปรไฟล์ใหม่ของผู้ใช้ใหม่ทันที เมื่อผู้ใช้เลือกภาพจากโดเรทอรีภายนอกระบบ โดยไม่มีการเปลี่ยนแปลงหน้าและการบันทึกลงในฐานข้อมูล เพื่อให้ผู้ใช้ได้ตรวจสอบความเหมาะสมก่อนยืนยันการอัปโหลดเพื่อเพิ่มไปยังโดเรทอรีภายนอกระบบ และบันทึกลงในฐานข้อมูล

2.4 PHP Data Objects

PHP Data Objects หรือ PDO คือส่วนเสริมของภาษาโปรแกรม PHP สำหรับการเข้าถึงฐานข้อมูลในภาษาโปรแกรม PHP ซึ่งมีความสามารถในการเชื่อมต่อกับฐานข้อมูลหลายประเภท ทำให้ผู้ใช้สามารถเรียกใช้คำสั่ง PDO ในการดำเนินการกับฐานข้อมูลอื่นได้หากมีการเปลี่ยนแปลงระบบจัดการฐานข้อมูล โดยฐานข้อมูลที่รองรับการใช้งาน PDO ได้แก่ MySQL PostgreSQL และ SQLite เป็นต้น

นอกจากนี้การใช้งาน PDO สามารถป้องกันการโจมตีผ่าน SQL Injection ได้ปลอดภัยขึ้น เนื่องจากการใช้ตัวแปรที่ไม่ถูกต้องในการเรียกใช้คำสั่ง execute จะป้องกันการทำงานของคำสั่ง ซึ่งลดความเสี่ยงที่อาจเกิดขึ้นกับฐานข้อมูลได้

โครงการได้นำ PDO มาใช้ในการดำเนินการกับฐานข้อมูลโดยเมื่อผู้ใช้เรียกฟังก์ชันที่ประกอบด้วยคำสั่งภาษา SQL โปรแกรมจะได้เรียกใช้งาน PDO โดยการ prepare คำสั่ง SQL จากนั้นจึง bind ตัวแปรใน query ด้วยข้อมูลที่ต้องการ จากนั้นจึงเรียก execute คำสั่งนั้นตามลำดับ

2.5 SESSION

SESSION หรือ `$_SESSION` คือตัวแปรประเภท associative array ซึ่งสามารถเก็บข้อมูลได้เหมือนกับตัวแปรทั่วไป แต่ตัวแปรนี้มีลักษณะเป็น superglobal ทำให้เมื่อมีการกำหนดค่าให้กับตัวแปรแล้ว ค่าของตัวแปรจะไม่สูญหายไปเมื่อมีการเปลี่ยนแปลงหน้า โดยก่อนเรียกใช้งานตัวแปร `$_SESSION` ผู้ใช้ต้องเรียกใช้ฟังก์ชัน `session_start()` และดำเนินการกำหนดค่าให้กับตัวแปร

Web session ถูกสร้างขึ้นมาเมื่อ browser ของผู้ใช้ติดต่อกับ URL ของเว็บไซต์และถูกทำลายเมื่อ client ปิดตัวลง และบันทึกข้อมูลลงใน session ของเซิร์ฟเวอร์ โดยหากต้องการลบข้อมูลในตัวแปรให้ทำการเรียกฟังก์ชัน `session_unset()` เพื่อลบข้อมูลบางอย่างออกจากตัวแปร หรือ `session_destroy()`; เพื่อลบข้อมูลทั้งหมดออกจาก session นั้น หรือการออกจากระบบ

ระบบของโครงการได้ทำการกำหนดค่าให้กับตัวแปร `$_SESSION` เมื่อมีการเข้าสู่ระบบเมื่อผู้ใช้ไปยังแต่ละหน้าของเว็บไซต์ เพื่อให้การแสดงผลมีความถูกต้อง และระบบจะไม่เปลี่ยนแปลงข้อมูลกระทั่งผู้ใช้ออกจากระบบ เว้นแต่ผู้ใช้ต้องการเปลี่ยนแปลงข้อมูลของตนเอง

บทที่ 3

การวิเคราะห์และออกแบบระบบ

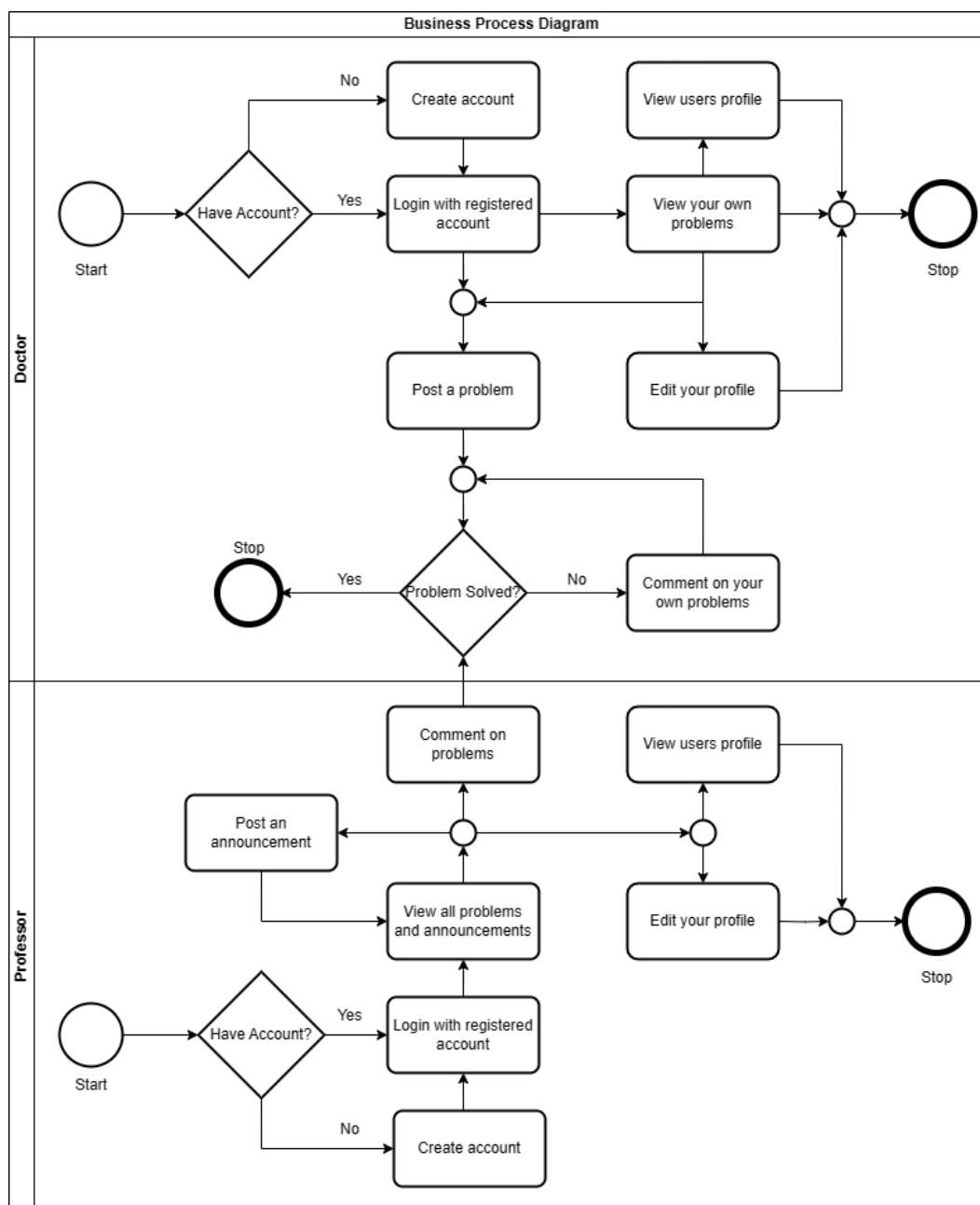
การวิเคราะห์และออกแบบระบบจะได้แสดงขั้นตอนการวิเคราะห์การทำงานของระบบตามลำดับการดำเนินงานของผู้ใช้ พร้อมทั้งแผนภาพประกอบ

3.1 การวิเคราะห์ระบบ

Disboard เป็นเว็บไซต์สำหรับการติดต่อสื่อสารในการดำเนินงานวิจัยติดต่อสื่อสารระหว่างอาจารย์และแพทย์นักวิจัย โดยแพทย์ผู้วิจัยสามารถโพสต์ปัญหางานวิจัยที่ต้องการความช่วยเหลือจากอาจารย์ และอาจารย์สามารถสนทนากับแพทย์เจ้าของปัญหานั้นได้ผ่านการแสดงความคิดเห็น โดยการออกแบบระบบจากกระบวนการดำเนินงานของระบบ คุณสมบัติของระบบ ขั้นตอนการดำเนินงานของแต่ละคุณสมบัติของระบบ เพื่อนำไปใช้ในการออกแบบฐานข้อมูล

3.2 กระบวนการดำเนินงานของระบบ

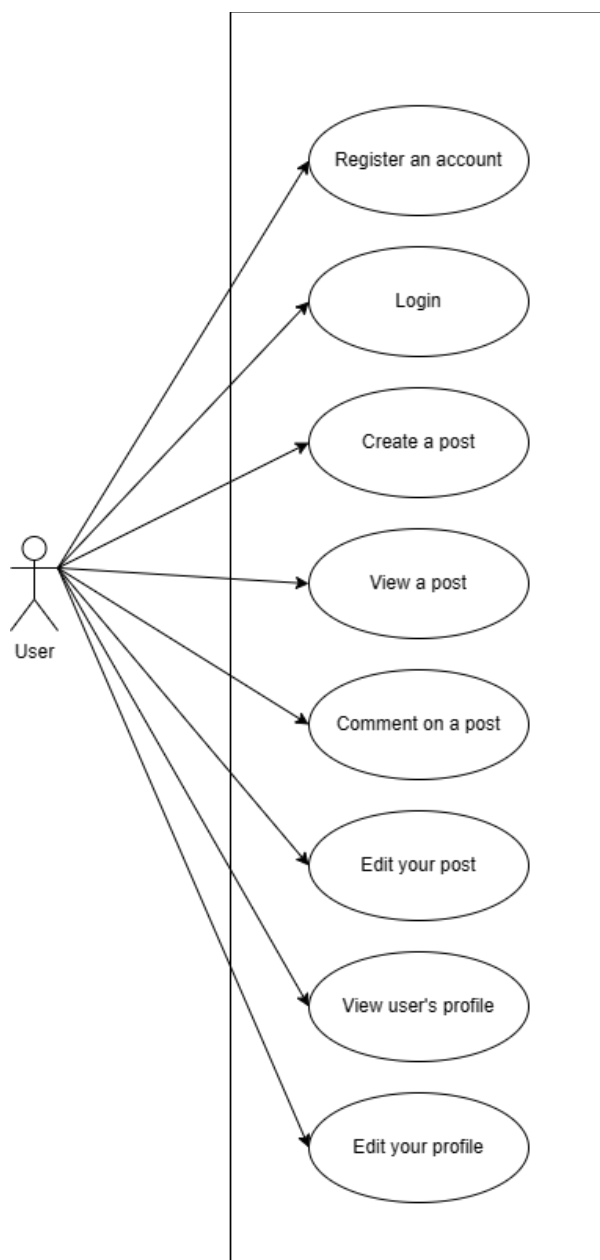
เมื่อแพทย์โพสต์ปัญหาทางงานวิจัยเพื่อขอความช่วยเหลือจากอาจารย์ อาจารย์ที่มีความสนใจสามารถตอบกลับโดยการแสดงความคิดเห็น หรือเมื่อแพทย์และอาจารย์ต้องการค้นหาผู้ที่มีความสนใจในงานวิจัยด้านเดียวกัน ผู้ใช้สามารถเรียกดูข้อมูลด้านวิชาชีพของผู้ใช้คนอื่นได้ ดังภาพ 3.1 ซึ่งจำแนกประเภทผู้ใช้ระหว่างแพทย์และอาจารย์ ในการระบุขั้นตอนการดำเนินงาน



ภาพที่ 3.1 แผนภาพกระบวนการดำเนินงานของระบบ

3.3 คุณสมบัติของระบบ

ระบบจำแนกประเภทของผู้ใช้ ได้แก่ แพทย์ และอาจารย์ โดยผู้ใช้ทั้ง 2 ประเภท สามารถสร้างบัญชีผู้ใช้ เข้าสู่ระบบ สร้างโพสต์ เข้าถึงโพสต์ แสดงความคิดเห็นบนโพสต์ แก้ไขโพสต์ของตนเอง แสดงข้อมูลในโปรไฟล์ของผู้ใช้แต่ละคน และแก้ไขข้อมูลในโปรไฟล์ของตนเอง ดังภาพที่ 3.2 แผนภาพระบุคุณสมบัติของระบบซึ่งผู้ใช้สามารถดำเนินการได้



ภาพที่ 3.2 แผนภาพคุณสมบัติของระบบ

3.4 ขั้นตอนการดำเนินงานของแต่ละคุณสมบัติของระบบ

ระบบนี้มีคุณสมบัติ 8 คุณสมบัติสำหรับผู้ใช้งานที่ได้กล่าวในบทที่ 3.3 โดยแต่ละคุณสมบัติมีขั้นตอนการดำเนินงานดังนี้

3.4.1 การสร้างบัญชีผู้ใช้

ระบบกำหนดให้ผู้ที่ต้องการใช้งานสร้างบัญชีผู้ใช้เพื่อเข้าถึงคุณสมบัติอื่นของระบบ โดยการกรอกข้อมูลลงในแบบฟอร์ม ซึ่งประกอบด้วย ชื่อและนามสกุล ชื่อบัญชีผู้ใช้ อีเมล รหัสผ่าน และภาพประจำบัญชีผู้ใช้หากผู้ใช้ต้องการ เมื่อผู้ใช้กรอกข้อมูลครบถ้วน จึงเลือกประเภทของสมาชิกระหว่างแพทย์หรืออาจารย์สำหรับการใช้งาน เมื่อระบบตรวจสอบข้อมูลสำเร็จ จะดำเนินการนำผู้ใช้ไปสู่คุณสมบัติอื่นถัดไป ดังตารางที่ 3.1 ซึ่งแสดงขั้นตอนการสร้างบัญชีผู้ใช้

UCID	1	
UC Name	Create an account	
Actor	User	
Pre-condition		
#	Actor	System
1	กรอกข้อมูลลงในแบบฟอร์ม	
2	เลือกประเภทสมาชิกของบัญชีผู้ใช้	
3		ตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้ทำการกรอกลงในแบบฟอร์ม
4		บันทึกข้อมูลลงในฐานข้อมูล
5		แสดงผลการสมัครสมาชิกสำเร็จ ดำเนินไปยังการเข้าสู่ระบบ
Alternative Course		หากผู้ใช้กรอกข้อมูลไม่ถูกต้อง แสดงข้อผิดพลาดเพื่อให้ผู้ใช้กรอกข้อมูลนั้นให้ถูกต้อง
Post-condition		

ตารางที่ 3.1 ขั้นตอนการสร้างบัญชีผู้ใช้

3.4.2 การเข้าสู่ระบบ

เมื่อผู้ที่มีบัญชีผู้ใช้จึงสามารถเข้าสู่ระบบได้ โดยระบบกำหนดให้ผู้ใช้งานกรอกข้อมูลได้แก่ ชื่อบัญชีผู้ใช้หรืออีเมล และรหัสผ่าน และเลือกเข้าสู่ระบบ โดยเมื่อระบบตรวจสอบความถูกต้องของข้อมูลสำเร็จ จึงนำผู้ใช้ไปยังส่วนการแสดงผลของบัญชีผู้ใช้นั้น เพื่อให้เลือกคุณสมบัติอื่นนอกเหนือจากการเข้าสู่ระบบถัดไป ดังตารางที่ 3.2 ซึ่งแสดงขั้นตอนการเข้าสู่ระบบ

UCID	2	
UC Name	Login	
Actor	User	
Pre-condition	Create an account	
#	Actor	System
1	กรอกข้อมูลลงในแบบฟอร์ม	
2	เลือกเข้าสู่ระบบ	
3		ตรวจสอบความถูกต้องของข้อมูลที่ใช้ทำการกรอกลงในแบบฟอร์ม
4		ดำเนินไปยังการแสดงผลโพสต์สำหรับบัญชีผู้ใช้นั้น
Alternative Course		หากผู้ใช้งานกรอกข้อมูลไม่ถูกต้องแสดงข้อผิดพลาดเพื่อให้ผู้ใช้งานกรอกข้อมูลนั้นให้ถูกต้อง
Post-condition		

ตารางที่ 3.2 ขั้นตอนการเข้าสู่ระบบ

3.4.3 การสร้างโพสต์

ผู้ใช้งานสามารถสร้างโพสต์ได้เมื่อเข้าสู่ระบบสำเร็จ โดยเลือกการสร้างโพสต์ และกรอกข้อมูลลงในแบบฟอร์ม ได้แก่ หัวข้อ รายละเอียด และเอกสารประกอบโพสต์หากต้องการ จากนั้นเลือกประเภทของโพสต์ เพื่อให้ระบบทำการตรวจสอบข้อมูลเพื่อบันทึกข้อมูลลงในฐานข้อมูล และบรรจุเอกสารลงในระบบ ดังตารางที่ 3.3 ซึ่งแสดงขั้นตอนการทำงานของระบบการสร้างโพสต์

UCID	3	
UC Name	Create a post	
Actor	User	
Pre-condition	Login	
#	Actor	System
1	เลือกสร้างโพสต์	
2	กรอกข้อมูลลงในแบบฟอร์ม	
3	เลือกประเภทของโพสต์	
4		ตรวจสอบความถูกต้องของข้อมูล ที่ใช้ทำการกรอกลงในแบบฟอร์ม
5		บรรจุเอกสารลงในระบบ
6		บันทึกข้อมูลของโพสต์ และข้อมูล ของเอกสารลงในฐานข้อมูล ดำเนิน ไปยังโพสต์ทั้งหมด
Alternative Course		หากผู้ใช้กรอกข้อมูลไม่ถูกต้อง แสดงข้อผิดพลาดเพื่อให้ผู้ใช้กรอก ข้อมูลนั้นให้ถูกต้อง หากการบรรจุเอกสารมีความ ผิดพลาด แสดงข้อผิดพลาดนั้นให้ ผู้ใช้ทราบ
Post-condition		

ตารางที่ 3.3 ขั้นตอนการสร้างโพสต์

3.4.4 การเข้าถึงโพสต์

ผู้ใช้งานสามารถเข้าถึงโพสต์ได้ โดยการเลือกหัวข้อโพสต์ที่ต้องการ หลังจากการเข้าสู่ระบบ โดยหากผู้ใช้งานมีประเภทสมาชิกเป็นอาจารย์ จะสามารถเลือกดูโพสต์ของทุกผู้ใช้ได้ แต่หากผู้ใช้งานมีประเภทสมาชิกเป็นแพทย์ สามารถเลือกดูได้เฉพาะโพสต์ปัญหาของตนเองเท่านั้น ดังตารางที่ 3.4 ซึ่งแสดงขั้นตอนการเข้าถึงโพสต์

UCID	4	
UC Name	View a post	
Actor	User	
Pre-condition	Login	
#	Actor	System
1	เลือกโพสต์ที่ต้องการ	
2		เลือกข้อมูลของโพสต์ เอกสาร และความคิดเห็นจากฐานข้อมูล
3		แสดงข้อมูลของโพสต์
Alternative Course		หากผู้ใช้งานกรอกที่อยู่ของโพสต์ที่ตนเองไม่มีสิทธิ์ในการเข้าถึง แสดงโพสต์ทั้งหมดของบัญชีผู้ใช้นั้น
Post-condition		

ตารางที่ 3.4 ขั้นตอนการเข้าถึงโพสต์

3.4.5 การแสดงความคิดเห็นบนโพสต์ปัญหาทางานวิจัย

ผู้ใช้งานสามารถแสดงความคิดเห็นบนโพสต์ปัญหาทางานวิจัย โดยอาจารย์สามารถแสดงความคิดเห็นบนโพสต์ปัญหาทางานวิจัยทุกโพสต์ แต่แพทย์สามารถแสดงความคิดเห็นบนโพสต์ปัญหาของตนเองได้เท่านั้น เมื่อผู้ใช้กรอกความคิดเห็น และเลือกแสดงความคิดเห็น ระบบจะทำการบันทึกข้อมูลความคิดเห็นลงในฐานข้อมูล และแสดงความคิดเห็นในโพสต์นั้น ดังตารางที่ 3.5 ซึ่งแสดงขั้นตอนการแสดงความคิดเห็นบนโพสต์

UCID	5	
UC Name	Comment on a post	
Actor	User	
Pre-condition	View a post	
#	Actor	System
1	เลือกโพสต์ที่ต้องการ	
2		เลือกข้อมูลของโพสต์ เอกสาร และความคิดเห็นจากฐานข้อมูล
3		แสดงส่วนการแสดงความคิดเห็นเมื่อประเภทของโพสต์คือปัญหา
4	กรอกความคิดเห็น	
5	เลือกแสดงความคิดเห็น	
6		ตรวจสอบความถูกต้องข้อมูล
7		บันทึกข้อมูลความคิดเห็นลงในฐานข้อมูล และอัปเดตข้อมูลเวลาของโพสต์
8		แสดงความคิดเห็นในโพสต์นั้น
Alternative Course		หากผู้ใช้ไม่กรอกความคิดเห็นระบบทำการแจ้งเตือนให้ผู้ใช้กรอกความคิดเห็น
Post-condition		

ตารางที่ 3.5 ขั้นตอนการแสดงความคิดเห็นบนโพสต์

3.4.6 การแก้ไขโพสต์ของผู้ใช้

ผู้ใช้งานสามารถแก้ไขโพสต์ของตนเองได้ โดยเลือกโพสต์ที่ต้องการ ระบบจะทำการตรวจสอบเจ้าของโพสต์ จากนั้นเลือกแก้ไขโพสต์ ระบบจะทำการแสดงข้อมูลโพสต์ปัจจุบันในรูปแบบฟอร์ม เมื่อผู้ใช้แก้ไขโพสต์สำเร็จ และเลือกเปลี่ยนแปลงข้อมูล ระบบจะบันทึกข้อมูลใหม่ของโพสต์ลงในฐานข้อมูล ดังตารางที่ 3.6 ซึ่งแสดงขั้นตอนการแก้ไขโพสต์ของผู้ใช้

UCID	6	
UC Name	Edit your post	
Actor	User	
Pre-condition	View a post	
#	Actor	System
1	เลือกโพสต์ที่ต้องการแก้ไข	
2		เลือกข้อมูลโพสต์จากฐานข้อมูล
3		แสดงข้อมูลโพสต์ปัจจุบัน
4	กรอกข้อมูลที่ต้องการแก้ไขลงในแบบฟอร์ม	
5		บันทึกข้อมูลใหม่ของโพสต์
3		แสดงผลข้อมูลโพสต์ที่เปลี่ยนแปลงสำเร็จ
Alternative Course		หากผู้ใช้กรอกข้อมูลไม่ถูกต้อง แสดงข้อผิดพลาดเพื่อให้ผู้ใช้กรอกข้อมูลนั้นให้ถูกต้อง
Post-condition		

ตารางที่ 3.6 ขั้นตอนการแก้ไขโพสต์ของผู้ใช้

3.4.7 การเข้าถึงข้อมูลในโปรไฟล์ของผู้ใช้

ผู้ใช้งานสามารถเข้าถึงโปรไฟล์ผู้ใช้งานแต่ละคนได้ โดยการเลือกการแสดงผลบัญชีผู้ใช้งานทั้งหมด จากนั้นเลือกชื่อบุคคลที่ต้องการ ระบบจะทำการแสดงผลข้อมูลของผู้ใช้ได้แก่ ชื่อและนามสกุล ภาพบัญชีผู้ใช้งาน หน่วยงานที่สังกัด ประวัติการศึกษา หัวข้อความสนใจ และลิงค์ไปยังเว็บไซต์ภายนอกของผู้ใช้นั้น ดังตารางที่ 3.7 ซึ่งแสดงขั้นตอนการเข้าถึงข้อมูลในโปรไฟล์ของผู้ใช้

UCID	7	
UC Name	View users' profile	
Actor	User	
Pre-condition	Login	
#	Actor	System
1	เลือกแสดงบัญชีผู้ใช้งานทั้งหมด	
2	เลือกบัญชีผู้ใช้งานที่ต้องการ	
3		เลือกข้อมูลของผู้ใช้ดังกล่าวจากฐานข้อมูล
4		แสดงผลข้อมูลของผู้ใช้
Alternative Course		หากระบบไม่พบบัญชีผู้ใช้งานจะดำเนินการไปยังการแสดงผลบัญชีผู้ใช้งานทั้งหมด
Post-condition		

ตารางที่ 3.7 ขั้นตอนการเข้าถึงข้อมูลในโปรไฟล์ของผู้ใช้

3.4.8 การแก้ไขข้อมูลในโปรไฟล์ของผู้ใช้

ผู้ใช้งานสามารถแก้ไขข้อมูลของตนเองได้ โดยเลือกบัญชีผู้ใช้งานของตนเอง และเลือกแก้ไขข้อมูล ระบบจะทำการแสดงข้อมูลของผู้ใช้ปัจจุบันในรูปแบบฟอร์ม เมื่อผู้ใช้แก้ไขข้อมูลสำเร็จ และเลือกเปลี่ยนแปลงข้อมูล ระบบจะบันทึกข้อมูลใหม่ของผู้ใช้ลงในฐานข้อมูล ดังตารางที่ 3.8 ซึ่งแสดงขั้นตอนการแก้ไขข้อมูลในโปรไฟล์ของผู้ใช้

UCID	8	
UC Name	Edit your profile	
Actor	User	
Pre-condition	View users' profile	
#	Actor	System
1	เลือกแสดงข้อมูลผู้ใช้งานของตนเอง	
2		เลือกข้อมูลผู้ใช้งานปัจจุบันจากฐานข้อมูล
3	กรอกข้อมูลที่ต้องการแก้ไขลงในแบบฟอร์ม	
4		บันทึกข้อมูลใหม่ลงในฐานข้อมูล
5		แสดงผลข้อมูลของผู้ใช้ที่เปลี่ยนแปลงสำเร็จ
Alternative Course		หากผู้ใช้กรอกข้อมูลไม่ถูกต้อง แสดงข้อผิดพลาดเพื่อให้ผู้ใช้กรอกข้อมูลนั้นให้ถูกต้อง
Post-condition		

ตาราง 3.8 ขั้นตอนการแก้ไขข้อมูลในโปรไฟล์ของผู้ใช้

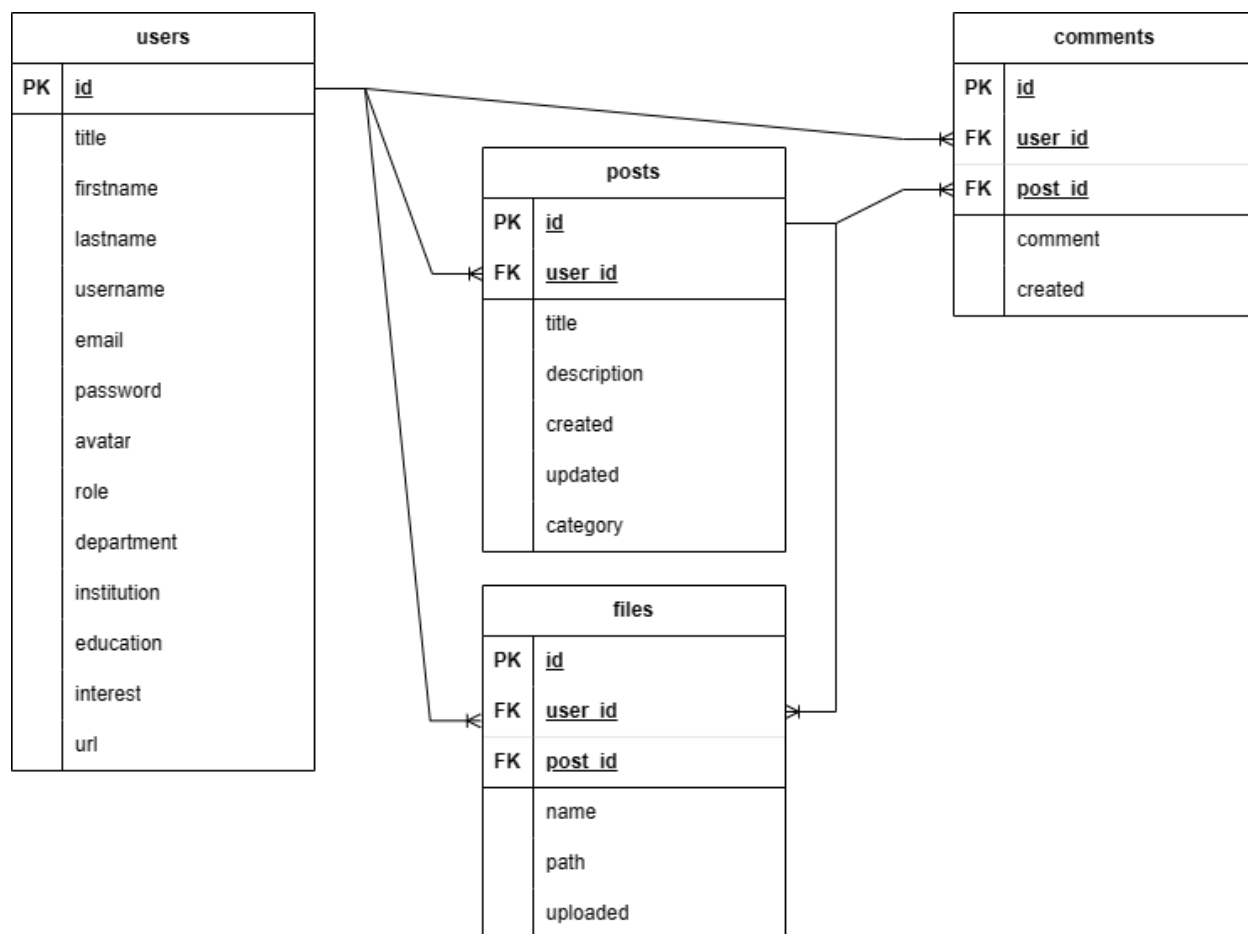
3.5 การออกแบบฐานข้อมูล

3.5.1 แผนภาพฐานข้อมูล

ฐานข้อมูลของโครงการประกอบด้วยตาราง 4 ตาราง ดังต่อไปนี้

1. ตาราง users จัดเก็บข้อมูลบัญชีผู้ใช้
2. ตาราง posts จัดเก็บข้อมูลโพสต์ของผู้ใช้
3. ตาราง files จัดเก็บข้อมูลเอกสารบนโพสต์ของผู้ใช้
4. ตาราง comments จัดเก็บข้อมูลความคิดเห็นของผู้ใช้ในโพสต์

โดยโครงสร้างของฐานข้อมูลแสดงได้ดังภาพที่ 3.1 แผนภาพโครงสร้างฐานข้อมูล

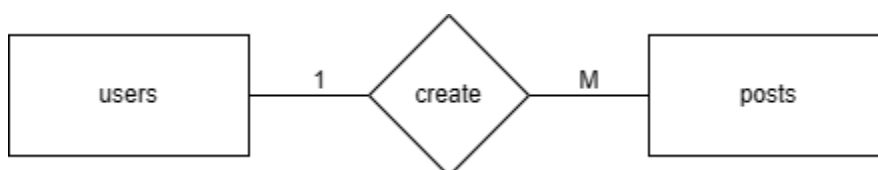


ภาพที่ 3.1 แผนภาพโครงสร้างของฐานข้อมูล

3.5.2 ความสัมพันธ์ระหว่างตารางของฐานข้อมูล

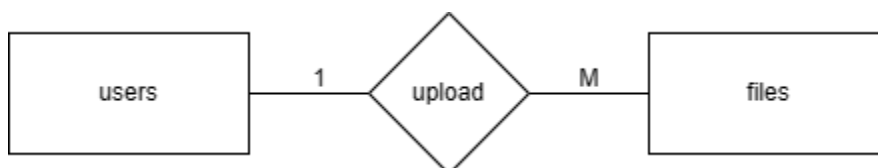
1. ตาราง users จัดเก็บข้อมูลบัญชีผู้ใช้ซึ่งประกอบด้วยข้อมูล id คำนำหน้าชื่อ ชื่อ นามสกุล ชื่อบัญชีผู้ใช้ อีเมล รหัสผ่าน ภาพประจำตัว ประเภทบัญชี ภาควิชา หน่วยงาน ประวัติการศึกษา ความสนใจ และเว็บไซต์ของผู้ใช้ โดยมีความสัมพันธ์กับตารางอื่นดังนี้

1.1 ตาราง posts จัดเก็บข้อมูลโพสต์ของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id หัวข้อโพสต์ รายละเอียด เวลาในการโพสต์ เวลาที่โพสต์มีการเปลี่ยนแปลง และประเภทของโพสต์ ซึ่งตาราง users มีความสัมพันธ์กับตาราง posts ในลักษณะ 1 to many ดังภาพที่ 3.2



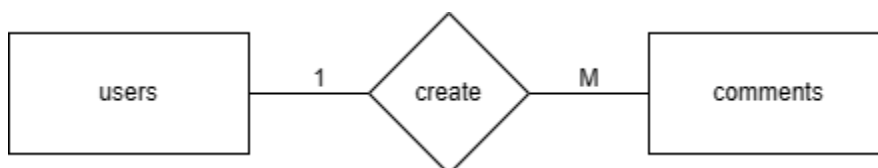
ภาพที่ 3.2 ความสัมพันธ์ระหว่างตาราง users และตาราง posts

1.2 ตาราง files จัดเก็บข้อมูลเอกสารในโพสต์ของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id, post id ชื่อเอกสาร ตำแหน่งในการจัดเก็บเอกสาร และเวลาในการอัปโหลดเอกสาร ซึ่งตาราง users มีความสัมพันธ์กับตาราง files ในลักษณะ 1 to many ดังภาพที่ 3.3



ภาพที่ 3.3 ความสัมพันธ์ระหว่างตาราง users และตาราง files

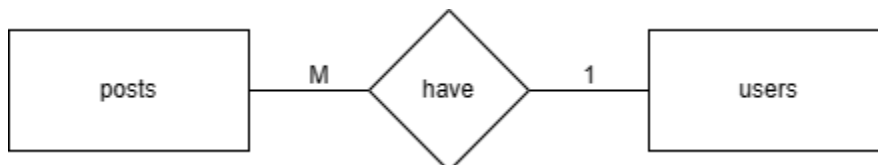
1.3 ตาราง comments จัดเก็บข้อมูลความคิดเห็นของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id, post id ความคิดเห็น และเวลาที่แสดงความคิดเห็น ซึ่งตาราง users มีความสัมพันธ์กับตาราง comments ในลักษณะ 1 to many ดังภาพที่ 3.4



ภาพที่ 3.4 ความสัมพันธ์ระหว่างตาราง users และตาราง posts

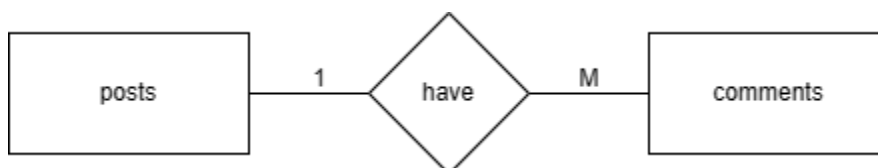
2. ตาราง posts จัดเก็บข้อมูลโพสต์ของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id หัวข้อโพสต์ รายละเอียดเวลาในการโพสต์ เวลาที่โพสต์มีการเปลี่ยนแปลง และประเภทของโพสต์ โดยมีความสัมพันธ์กับตารางอื่นดังนี้

2.1 ตาราง users จัดเก็บข้อมูลบัญชีผู้ใช้ซึ่งประกอบด้วยข้อมูล id คำนามหน้าชื่อ ชื่อ นามสกุล ชื่อบัญชีผู้ใช้อีเมล รหัสผ่าน ภาพประจำตัว ประเภทบัญชี ภาควิชา หน่วยงาน ประวัติการศึกษา ความสนใจ และเว็บไซต์ของผู้ใช้ ซึ่งตาราง posts มีความสัมพันธ์กับตาราง users ในลักษณะ Many to 1 ดังภาพที่ 3.5



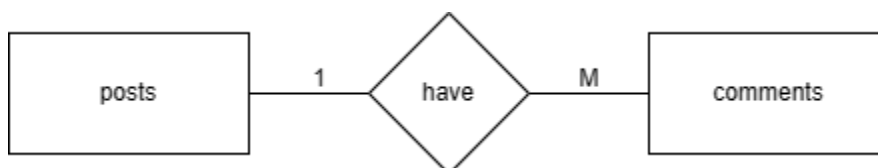
ภาพที่ 3.5 ความสัมพันธ์ระหว่างตาราง posts และตาราง users

2.2 ตาราง files จัดเก็บข้อมูลเอกสารในโพสต์ของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id, post id ชื่อเอกสาร ตำแหน่งในการจัดเก็บเอกสาร และเวลาในการอัปโหลดเอกสาร ซึ่งตาราง posts มีความสัมพันธ์กับตาราง files ในลักษณะ 1 to many ดังภาพที่ 3.6



ภาพที่ 3.6 ความสัมพันธ์ระหว่างตาราง posts และตาราง files

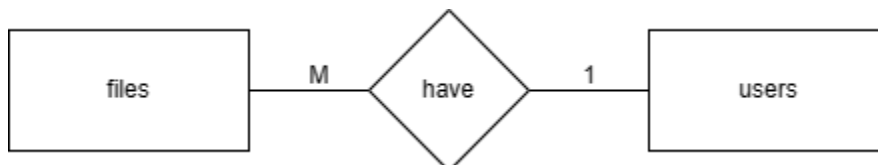
2.3 ตาราง comments จัดเก็บข้อมูลความคิดเห็นของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id, post id ความคิดเห็น และเวลาที่แสดงความคิดเห็น ซึ่งตาราง posts มีความสัมพันธ์กับตาราง comments ในลักษณะ 1 to many ดังภาพที่ 3.7



ภาพที่ 3.7 ความสัมพันธ์ระหว่างตาราง posts และตาราง comments

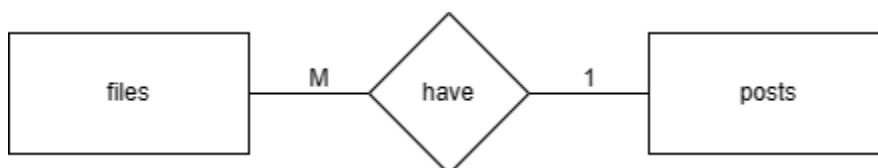
3. ตาราง files จัดเก็บข้อมูลเอกสารในโพสต์ของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id, post id ชื่อเอกสาร ตำแหน่งในการจัดเก็บเอกสาร และเวลาในการอัปโหลดเอกสาร โดยมีความสัมพันธ์กับตารางอื่นดังนี้

3.1 ตาราง users จัดเก็บข้อมูลบัญชีผู้ใช้ซึ่งประกอบด้วยข้อมูล id คำนำหน้าชื่อ ชื่อ นามสกุล ชื่อบัญชีผู้ใช้อีเมล รหัสผ่าน ภาพประจำตัว ประเภทบัญชี ภาควิชา หน่วยงาน ประวัติการศึกษา ความสนใจ และเว็บไซต์ของผู้ใช้ ซึ่งตาราง files มีความสัมพันธ์กับตาราง users ในลักษณะ Many to 1 ดังภาพที่ 3.8



ภาพที่ 3.8 ความสัมพันธ์ระหว่างตาราง files และตาราง users

3.2 ตาราง posts จัดเก็บข้อมูลโพสต์ของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id หัวข้อโพสต์ รายละเอียด เวลาในการโพสต์ เวลาที่โพสต์มีการเปลี่ยนแปลง และประเภทของโพสต์ ซึ่งตาราง files มีความสัมพันธ์กับตาราง posts ในลักษณะ Many to 1 ดังภาพที่ 3.9



ภาพที่ 3.9 ความสัมพันธ์ระหว่างตาราง files และตาราง posts

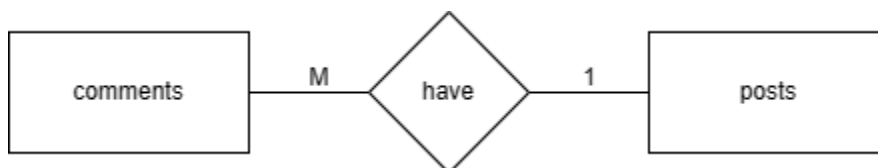
4. ตาราง comments จัดเก็บข้อมูลความคิดเห็นของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id, post id ความคิดเห็น และเวลาที่แสดงความคิดเห็น โดยมีความสัมพันธ์กับตารางอื่นดังนี้

4.1. ตาราง users จัดเก็บข้อมูลบัญชีผู้ใช้ซึ่งประกอบด้วยข้อมูล id คำนำหน้าชื่อ ชื่อ นามสกุล ชื่อบัญชีผู้ใช้อีเมล รหัสผ่าน ภาพประจำตัว ประเภทบัญชี ภาควิชา หน่วยงาน ประวัติการศึกษา ความสนใจ และเว็บไซต์ของผู้ใช้ ซึ่งตาราง comments มีความสัมพันธ์กับตาราง users ในลักษณะ Many to 1 ดังภาพที่ 3.10



ภาพที่ 3.10 ความสัมพันธ์ระหว่างตาราง comments และตาราง users

4.2 ตาราง posts จัดเก็บข้อมูลโพสต์ของผู้ใช้ซึ่งประกอบด้วยข้อมูล id, user id หัวข้อโพสต์ รายละเอียด เวลาในการโพสต์ เวลาที่โพสต์มีการเปลี่ยนแปลง และประเภทของโพสต์ ซึ่งตาราง comments มีความสัมพันธ์กับตาราง posts ในลักษณะ Many to 1 ดังภาพที่ 3.11



ภาพที่ 3.11 ความสัมพันธ์ระหว่างตาราง comments และตาราง posts

บทที่ 4

การพัฒนาระบบ

การพัฒนาระบบตามการวิเคราะห์และออกแบบระบบนี้ ระบุถึงวิธีการพัฒนาเว็บไซต์ด้วยภาษาโปรแกรม PHP ในการพัฒนาระบบการเชื่อมต่อไปยังฐานข้อมูล ร่วมกับภาษา HTML ในการแสดงผลข้อมูล โดยลำดับตามขั้นตอนการใช้งาน ดังนี้ การเข้าและออกจากระบบ การโพสต์ การแสดงความคิดเห็นบนโพสต์ปัญหาทางวิจัย และการเข้าถึงและแก้ไขโปรไฟล์

4.1 การเข้าและออกจากระบบ

การเข้าสู่ระบบเพื่อให้ผู้ใช้สามารถเข้าถึงระบบอื่น ประกอบด้วยการสร้างบัญชีผู้ใช้ การเข้าสู่ระบบ รวมถึงการออกจากระบบเมื่อสิ้นสุดการใช้งาน

4.1.1 การสร้างบัญชีผู้ใช้

1. การกำหนดค่าตัวแปร

พารามิเตอร์ในอาเรย์ \$_POST จะรับข้อมูลที่ถูกส่งมาจากการสร้างบัญชีผู้ใช้ และถูกกรองอักขระพิเศษผ่านฟังก์ชัน filter_var เพื่อความปลอดภัย และกำหนดค่าให้กับตัวแปร

2. การตรวจสอบข้อมูล

ตรวจสอบความถูกต้องของข้อมูลในตัวแปร ได้แก่ ข้อมูลไม่เป็นค่าว่าง พบ username และ email ในระบบ และมีการอัปโหลดภาพโปรไฟล์หรือไม่ หากไม่มีจะใช้ภาพเริ่มต้น

3. การดำเนินการของระบบ

3.1 การนำภาพโปรไฟล์เข้าสู่ไดเรกทอรีในระบบด้วยฟังก์ชัน

```
if (!move_uploaded_file($tmp_name, $avatar_destination_path))
```

3.2 เรียกใช้คำสั่ง

```
INSERT INTO users (title, firstname, lastname, username, email, password, role, avatar) VALUES (:title, :firstname, :lastname, :username, :email, :password, :role, :avatar)
```

ในการบันทึกข้อมูล

3.3 เมื่อบันทึกข้อมูลสำเร็จ เปลี่ยนหน้าไปยังหน้าการเข้าสู่ระบบ

4.1.2 การเข้าสู่ระบบ

1. การกำหนดค่าตัวแปร

พารามิเตอร์ในอาเรย์ \$_POST จะรับข้อมูลที่ถูกส่งมาจากการเข้าสู่ระบบ และถูกกรองอักขระพิเศษผ่านฟังก์ชัน filter_var เพื่อความปลอดภัย และกำหนดค่าให้กับตัวแปร

2. การตรวจสอบข้อมูล

การเรียกใช้คำสั่ง `SELECT * FROM users WHERE username = :username OR email = :email` ในการค้นหาบัญชีผู้ใช้

3. การดำเนินการของระบบ

3.1 การตรวจสอบรหัสผ่านของบัญชีผู้ใช้ด้วยฟังก์ชัน

```
if(password_verify($password, $user['password']))
```

3.2 กำหนดข้อมูลของผู้ใช้ปัจจุบันในและไปยังหน้าโพสต์ของผู้ใช้ด้วย

```
$_SESSION['user'] = $user;
header('Location: ../pages/index.php');
exit();
```

4.1.3 การออกจากระบบ

1. การดำเนินการของระบบ

1.1 ลบข้อมูลผู้ใช้ออกจากอาเรย์ \$_SESSION ด้วยฟังก์ชัน session_destroy();

1.2 เปลี่ยนหน้าไปยังหน้าการเข้าสู่ระบบ

4.2 การโพสต์

ผู้ใช้เลือกสร้างโพสต์ปัญหาทางวิจัย หรือประกาศข่าวสาร หากผู้ใช้เป็นอาจารย์ โดยผู้ใช้สามารถแก้ไขโพสต์ของตนเอง และเข้าถึงโพสต์ที่มีสิทธิ์เท่านั้น

4.2.1 การสร้างโพสต์

1. การกำหนดค่าตัวแปร

พารามิเตอร์ในอาเรย์ \$_POST จะรับข้อมูลที่ถูกส่งมาจากการสร้างโพสต์ และถูกกรองอักขระพิเศษผ่านฟังก์ชัน filter_var เพื่อความปลอดภัย และกำหนดค่าให้กับตัวแปร

2. การตรวจสอบข้อมูล

ตรวจสอบความถูกต้องของข้อมูล ได้แก่ หัวเรื่องหรือรายละเอียดมีค่าว่าง และการอัปโหลดเอกสาร

3. การดำเนินการของระบบ

3.1 เรียกใช้คำสั่ง INSERT INTO posts (title, description, user_id, category) VALUES (:title, :description, :user_id, :category) บันทึกข้อมูลโพสต์ลงในฐานข้อมูล

3.2 ระบุ id ของโพสต์สำหรับการเพิ่มไฟล์ด้วย

```
$post_id = $connection->lastInsertId();
```

3.3 ตรวจสอบข้อมูลของเอกสารแต่ละไฟล์ และนำเข้าสู่ไดเรกทอรีด้วย

```
foreach ($files['tmp_name'] as $key => $tmp_name)
    if(!move_uploaded_file($tmp_name, $file_destination_path))
```

3.4 เรียกใช้คำสั่ง INSERT INTO files (post_id, user_id, name, path) VALUES (:post_id, :user_id, :name, :path) บันทึกข้อมูลเอกสารลงในฐานข้อมูล

3.5 เปลี่ยนหน้าไปยังหน้าโพสต์ของผู้ใช้

4.2.2 การแก้ไขโพสต์

1. การกำหนดค่าตัวแปร

พารามิเตอร์ในอาเรย์ \$_POST จะรับข้อมูลที่ถูกส่งมาจากการแก้ไขโพสต์ และถูกกรองอักขระพิเศษผ่านฟังก์ชัน filter_var เพื่อความปลอดภัย และกำหนดค่าให้กับตัวแปร

2. การตรวจสอบข้อมูล

ตรวจสอบความถูกต้องของข้อมูล ได้แก่ หัวเรื่องหรือรายละเอียดมีค่าว่าง และการอัปโหลดเอกสาร และบันทึกเวลาแก้ไขโพสต์ด้วย `$datetime = date('Y-m-d H:i:s');`

3. การดำเนินการของระบบ

3.1 เรียกใช้คำสั่ง `UPDATE posts SET title = :title, description = :description, updated = :updated WHERE id = :id` เปลี่ยนแปลงข้อมูลของโพสต์ในฐานข้อมูล

3.2 เรียกใช้คำสั่ง `SELECT path FROM files WHERE post_id = :post_id` เรียกตำแหน่งของเอกสารของโพสต์นั้น

3.3 ลบเอกสารเดิมออกจากไดเรกทอรีด้วย

```
if (file_exists($path))
    unlink($path);
```

3.4 เรียกใช้คำสั่ง `DELETE FROM files WHERE post_id = :post_id` ลบข้อมูลเอกสารออกจากฐานข้อมูล

3.5 ตรวจสอบข้อมูลของเอกสารใหม่แต่ละไฟล์ และนำเข้าสู่ไดเรกทอรีด้วย

```
foreach ($files['tmp_name'] as $key => $tmp_name)
    if(!move_uploaded_file($tmp_name, $file_destination_path))
```

3.6 เรียกใช้คำสั่ง `INSERT INTO files (post_id, user_id, name, path) VALUES (:post_id, :user_id, :name, :path)` บันทึกข้อมูลเอกสารใหม่ลงในฐานข้อมูล

3.7 เปลี่ยนหน้าไปยังหน้าโพสต์ที่ผ่านการแก้ไขสำเร็จ

4.2.3 การเข้าถึงโพสต์

1. การกำหนดค่าตัวแปร

พารามิเตอร์ในอาร์เรย์ `$_GET` จะรับข้อมูล id ของโพสต์ที่ถูกส่งมาจากหน้าแสดงโพสต์ทั้งหมด

2. การตรวจสอบข้อมูล

การเรียกใช้คำสั่ง `SELECT posts.*, users.username, users.avatar, users.firstname, users.lastname FROM posts JOIN users ON posts.user_id = users.id WHERE posts.id = :post_id` เรียกข้อมูลโพสต์ ข้อมูลชื่อบัญชี ภาพโปรไฟล์ ชื่อ และนามสกุลของผู้โพสต์

3. การดำเนินการของระบบ

3.1 เรียกใช้คำสั่ง `SELECT * FROM files WHERE post_id = :post_id` เรียกข้อมูลเอกสารของโพสต์นั้น

3.2 เรียกใช้คำสั่ง `SELECT comments.*, users.avatar, users.firstname, users.lastname, users.role FROM comments JOIN users ON comments.user_id = users.id WHERE comments.post_id = :post_id` เรียกข้อมูลความคิดเห็นของโพสต์นั้น

3.3 กำหนดค่าในตัวแปร `$_SESSION['post_id']` มีค่า id ของโพสต์ดังกล่าว เปลี่ยนหน้าไปยังโพสต์นั้น และแสดงข้อมูลผ่านฟังก์ชัน `nl2br` ในการแสดงบรรทัดและช่องว่าง

4.3 การแสดงความคิดเห็น

ผู้ใช้งานสามารถแสดงความคิดเห็นบนโพสต์ประเภทปัญหาทางวิจัย โดยแพทย์สามารถแสดงความคิดเห็นบนโพสต์ของตนเองเท่านั้น

1. การกำหนดค่าตัวแปร

พารามิเตอร์ในอาเรย์ `$_POST` จะรับข้อมูลที่ถูกส่งมาจากแบบฟอร์มการแสดงความคิดเห็น และถูกกรองอักขระพิเศษผ่านฟังก์ชัน `filter_var` เพื่อความปลอดภัย และกำหนดค่าให้กับตัวแปร

2. การตรวจสอบข้อมูล

ตรวจสอบความถูกต้องของข้อมูล ได้แก่ ความคิดเห็นมีค่าว่าง

3. การดำเนินการของระบบ

3.1 เรียกใช้คำสั่ง INSERT INTO posts (title, description, user_id, category) VALUES (:title, :description, :user_id, :category) บันทึกข้อมูลโพสต์ลงในฐานข้อมูล

3.2 ระบุ id ของโพสต์สำหรับการเพิ่มไฟล์ด้วย

```
$post_id = $connection->lastInsertId();
```

3.3 ตรวจสอบข้อมูลของเอกสารแต่ละไฟล์ และนำเข้าสู่ไดเรกทอรีด้วย

```
foreach ($files['tmp_name'] as $key => $tmp_name)
    if(!move_uploaded_file($tmp_name, $file_destination_path))
```

3.4 เรียกใช้คำสั่ง INSERT INTO files (post_id, user_id, name, path) VALUES (:post_id, :user_id, :name, :path) บันทึกข้อมูลเอกสารลงในฐานข้อมูล

3.5 เปลี่ยนหน้าไปยังหน้าโพสต์ของผู้ใช้ และแสดงความคิดเห็นใหม่ผ่านฟังก์ชัน nl2br ในการแสดงบรรทัดและช่องว่าง

4.4 การเข้าถึงและแก้ไขโปรไฟล์

ผู้ใช้สามารถเข้าถึงหน้าโปรไฟล์ของผู้ใช้อื่นในการเรียกดูข้อมูล และแก้ไขข้อมูลในโปรไฟล์ของตนเองได้

4.4.1 การเข้าถึงโปรไฟล์

1. การกำหนดค่าตัวแปร

พารามิเตอร์ในอาเรย์ \$_GET จะรับข้อมูล id ของผู้ใช้ซึ่งถูกส่งมาจากหน้าแสดงผู้ใช้ทั้งหมด

2. การตรวจสอบข้อมูล

การเรียกใช้คำสั่ง SELECT * FROM users WHERE users.id = :id เรียกข้อมูลผู้ใช้

3. การดำเนินการของระบบ

3.1 ตรวจสอบความเป็นเจ้าของโปรไฟล์ของผู้ใช้ด้วย

```
if ($_SESSION['user']['id'] == $user['id'])
```

3.2 แสดงข้อมูลของผู้ใช้นั้นผ่านฟังก์ชัน nl2br ในการแสดงช่องว่าง บรรทัด และอักขระพิเศษ

4.4.2 การแก้ไขโปรไฟล์

1. การกำหนดค่าตัวแปร

พารามิเตอร์ในอาร์เรย์ `$_POST` จะรับข้อมูลที่ถูกส่งมาจากการแก้ไขโปรไฟล์ และถูกกรองอักขระพิเศษผ่านฟังก์ชัน `filter_var` เพื่อความปลอดภัย และกำหนดค่าให้กับตัวแปร

2. การตรวจสอบข้อมูล

ตรวจสอบความถูกต้องของข้อมูล ได้แก่ ชื่อและนามสกุลไม่เป็นค่าว่าง และมีการเปลี่ยนภาพโปรไฟล์ใหม่

3. การดำเนินการของระบบ

3.1 ลบภาพเดิมออกจากไดเรกทอรีด้วย `unlink($avatar_destination_path);`

3.2 เรียกใช้คำสั่ง

```
UPDATE users SET title = :title,firstname = :firstname,
lastname = :lastname, department = :department,
institution = :institution, education = :education,
interests = :interests, avatar = :avatar, url = :url WHERE id = :id
```

เปลี่ยนแปลงข้อมูลผู้ใช้ในฐานข้อมูล

3.3 กำหนดข้อมูลที่ผ่านการแก้ไขให้กับพารามิเตอร์ในตัวแปร `$_SESSION['user']` ด้วย

```
$_SESSION['user'] = ['title' => $title, 'firstname' => $firstname,
'lastname' => $lastname, 'department' => $department,
'institution' => $institution, 'education' => $education,
'interests' => $interests, 'url' => $url,] + $_SESSION['user'];
```

3.4 เปลี่ยนหน้าไปยังหน้าโปรไฟล์ของผู้ใช้ปัจจุบันจากตัวแปร `$_SESSION['user']['id']`

บทที่ 5

การทดสอบระบบ

การทดสอบระบบดำเนินการโดยการเชื่อมต่อกับฐานข้อมูล และสร้างตารางฐานข้อมูล จากนั้นจึงเริ่มต้นระบบเพื่อเริ่มการใช้งานเว็บไซต์

5.1 การเชื่อมต่อฐานข้อมูล

เมื่อเข้าถึงเว็บไซต์ผ่านเว็บเบราว์เซอร์ ระบบจะทำการเริ่มต้นเซสชันของผู้ใช้ด้วยฟังก์ชัน `session_start()`; จากนั้นกำหนดเขตเวลาในระบบเป็นประเทศไทย และกำหนดตัวแปรในการเชื่อมต่อฐานข้อมูลได้แก่

1. `$host` ที่ตั้งของเว็บไซต์ มีค่า `localhost` ตำแหน่งเครื่องคอมพิวเตอร์ปัจจุบัน
2. `$name` ชื่อฐานข้อมูล มีค่า `blog` เพื่อสร้างตารางในฐานข้อมูล
3. `$user` ชื่อผู้ใช้ในการยืนยันตัวตนเข้าถึงฐานข้อมูล มีค่า `root` ในการเปลี่ยนแปลงข้อมูลโดยตรง
4. `$password` รหัสผ่านในการยืนยันตัวตนเข้าถึงฐานข้อมูล

โดยกำหนด Data Source ซึ่งมีพารามิเตอร์ได้แก่ตัวแปร `$host` และ `$name` และกำหนดตัวเลือกในการเชื่อมต่อฐานข้อมูลของ PHP Data Objects ดังนี้

1. การรายงานข้อผิดพลาด ทำการแสดงข้อผิดพลาดเมื่อการดำเนินการกับฐานข้อมูลมีความผิดพลาด
2. การเรียกข้อมูลจากฐานข้อมูล โดยคืนค่าข้อมูลเป็นอาร์เรย์ associative เพื่อเข้าถึงผ่าน index
3. การกำหนดการเตรียมการในการดำเนินการกับฐานข้อมูล โดยมีค่าเป็น `false` ซึ่งกำหนดให้มีการใช้ `prepare statement` จริงในการติดต่อฐานข้อมูล แทนการจำลองการทำงาน เพื่อเพิ่มความปลอดภัย

จากนั้นทำการเชื่อมต่อไปยังฐานข้อมูลโดย PHP Data Objects ด้วย

```
$connection = new PDO($dsn, $user, $password, $opt);
```

หากการเชื่อมต่อล้มเหลว แสดงข้อผิดพลาดทางจอภาพ โดยการทำงานของ การเชื่อมต่อฐานข้อมูลนี้จะเรียกใช้ในทุกหน้าของระบบ

5.2 การสร้างตารางฐานข้อมูล

เมื่อเชื่อมต่อไปยังฐานข้อมูลครั้งแรก ทำการสร้างตารางบนฐานข้อมูลนั้น โดยประกอบไปด้วย 4 ตาราง

5.2.1 ตาราง users

ตารางสำหรับจัดเก็บข้อมูลผู้ใช้โดยระบุแต่ละเรคอร์ดด้วย id และระบุข้อมูลที่จำเป็นในการสร้างบัญชีผู้ใช้ให้ไม่สามารถไม่มีค่า และข้อมูลในหน้าแสดงโปรไฟล์ไม่มีค่า หรือมีค่าว่างสำหรับข้อมูลประเภทข้อความ โดยเรียกคำสั่ง

```
CREATE TABLE users (
    `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    `title` varchar(50), `firstname` varchar(50) NOT NULL,
    `lastname` varchar(50) NOT NULL, `username` varchar(50) NOT NULL,
    `email` varchar(100) NOT NULL, `password` varchar(255) NOT NULL,
    `avatar` varchar(255) NOT NULL,
    `role` ENUM('doctor','professor','admin') NOT NULL,
    `department` varchar(255) DEFAULT NULL,
    `institution` varchar(255) DEFAULT NULL,
    `education` text, `interests` text,
    `url` varchar(255) DEFAULT NULL);
```

5.2.2 ตาราง posts

ตารางสำหรับจัดเก็บข้อมูลโพสต์โดยระบุแต่ละเรคอร์ดด้วย id และระบุข้อมูลทั้งหมดให้ไม่สามารถไม่มีค่า กำหนดข้อมูลเวลาโดยฟังก์ชัน current_timestamp และเชื่อมต่อกับตาราง users ผ่าน user id โดยเรียกคำสั่ง

```
CREATE TABLE posts (
    `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    `title` varchar(255) NOT NULL, `description` text NOT NULL,
    `created` DATETIME NOT NULL DEFAULT current_timestamp(),
    `updated` DATETIME NOT NULL DEFAULT current_timestamp(),
    `user_id` int(11) UNSIGNED NOT NULL,
    `category` ENUM('problem', 'announcement') NOT NULL,
    FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
    ON DELETE CASCADE);
```

5.2.3 ตาราง files

ตารางสำหรับจัดเก็บข้อมูลเอกสารของโพสต์โดยระบุแต่ละเรคอร์ดด้วย id และระบุข้อมูลทั้งหมดให้ไม่สามารถไม่มีค่า กำหนดข้อมูลเวลาโดยฟังก์ชัน current_timestamp เชื่อมต่อกับตาราง posts ผ่าน post id และตาราง users ผ่าน user id โดยเรียกคำสั่ง

```
CREATE TABLE files (
    `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    `post_id` int(11) UNSIGNED NOT NULL,
    `user_id` int(11) UNSIGNED NOT NULL,
    `name` VARCHAR(255) NOT NULL, `path` VARCHAR(1023) NOT NULL,
    `uploaded` DATETIME NOT NULL DEFAULT current_timestamp(),
    FOREIGN KEY (`post_id`) REFERENCES `posts` (`id`)
    ON DELETE CASCADE,
    FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
    ON DELETE CASCADE);
```


5.2.4 ตาราง comments

ตารางสำหรับจัดเก็บข้อมูลความคิดเห็นของโพสต์โดยระบุแต่ละเรคอร์ดด้วย id และระบุข้อมูลทั้งหมดให้ไม่สามารถไม่มีค่า กำหนดข้อมูลเวลาโดยฟังก์ชัน current_timestamp เชื่อมต่อกับตาราง posts ผ่าน post id และตาราง users ผ่าน user id โดยเรียกคำสั่ง

```
CREATE TABLE comments (
    `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    `user_id` int(11) UNSIGNED NOT NULL,
    `post_id` int(11) UNSIGNED NOT NULL,
    `comment` text NOT NULL,
    `created` DATETIME NOT NULL DEFAULT current_timestamp(),
    FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
    ON DELETE CASCADE,
    FOREIGN KEY (`post_id`) REFERENCES `posts` (`id`)
    ON DELETE CASCADE);
```

5.3 การเริ่มต้นระบบ

ระบบมีการทำงานผ่าน service 2 โปรแกรม ได้แก่ Web Server แสดงเว็บไซต์ และ Database Server จัดการฐานข้อมูล

5.3.1 Web Server

Docker ทำการสร้างอิมเมจ โดยการ mount ไดเรกทอรี src ลงใน /var/www/html บน container เชื่อมต่อพอร์ต 9000 และพอร์ต 80 ของ container และกำหนดให้ Database server เริ่มต้นก่อน Web Server ในการสร้างฐานข้อมูล รวมถึงกำหนดสภาพแวดล้อมให้กับ Web Server

5.3.2 Database Server

Docker ใช้อิมเมจ mariadb สำหรับฐานข้อมูลโดยกำหนดผู้ใช้ที่สร้าง container และกำหนดให้เข้าถึงฐานข้อมูลผ่าน service ของ phpmyadmin รวมถึงกำหนดสภาพแวดล้อมให้เชื่อมต่อกับ Database Server

บทที่ 6

บทสรุป ปัญหา ข้อเสนอแนะ และแนวทางการพัฒนาต่อยอด

จากการดำเนินการโครงการสามารถสรุปได้จากวัตถุประสงค์ของโครงการ โดยพบปัญหาระหว่างการดำเนินงานจำนวนหนึ่ง จึงมีข้อเสนอแนะ เพื่อนำไปสู่แนวทางการพัฒนาต่อยอดต่อไป

6.1 บทสรุป

โครงการนี้มีวัตถุประสงค์ให้อาจารย์สามารถติดต่อสื่อสารกับแพทย์ในการให้ความช่วยเหลือในการดำเนินงานวิจัย ผ่านการโพสต์และแสดงความคิดเห็น เพื่อให้แพทย์สามารถดำเนินงานวิจัยได้อย่างมีประสิทธิภาพ รวมถึงสามารถค้นหาผู้ที่มีความสนใจการดำเนินงานวิจัยร่วมกันได้จากการเข้าถึงโปรไฟล์ของผู้ใช้แต่ละคน ซึ่งช่วยอำนวยความสะดวกให้กับแพทย์และอาจารย์ในการติดต่อสื่อสาร ลดอุปสรรคระหว่างการดำเนินงานจากการที่อาจารย์ไม่ได้รับข้อมูลปัญหงานวิจัยที่มีประสิทธิภาพได้

6.2 ปัญหา

การดำเนินการโครงการได้พบปัญหาในแต่ละขั้นตอน ซึ่งส่งผลให้การพัฒนาระบบเกิดความล่าช้าดังนี้

1. การเลือกใช้ภาษาโปรแกรมที่ไม่เหมาะสมต่อความสามารถในการพัฒนาเว็บไซต์
2. การวิเคราะห์และออกแบบระบบที่ไม่ถูกต้องและเหมาะสมกับการใช้งานของผู้ใช้
3. การพัฒนาระบบที่เปลี่ยนแปลงรูปแบบหลายครั้ง ส่งผลให้เว็บไซต์มีการทำงานที่ไม่สมบูรณ์
4. การจำกัดความสามารถในการใช้งานของผู้ใช้แต่ละประเภทที่ไม่เหมาะสม เช่น แพทย์ไม่สามารถแก้ไขข้อมูลผู้ใช้ของตนเอง หรืออาจารย์ไม่สามารถดูโพสต์ของอาจารย์ท่านอื่นได้ เป็นต้น
5. การวางแผนการดำเนินงานที่ไม่มีประสิทธิภาพ ส่งผลให้คุณสมบัติของระบบบางอย่างถูกจำกัดลง

6.3 ข้อเสนอแนะ

เพื่อให้การดำเนินการของโครงการมีประสิทธิภาพ ควรปรึกษาผู้อื่นระหว่างการทำงานร่วม รวมถึงดำเนินงานอย่างสม่ำเสมอเป็นประจำ เพื่อให้โครงการมีความคืบหน้าตลอดเวลา และวางแผนในแต่ละขั้นตอนให้เหมาะสมด้วย

6.4 แนวทางการพัฒนาต่อยอด

1. เพิ่มระบบแจ้งเตือนผู้ใช้เมื่อมีการสร้างโพสต์ การแสดงความคิดเห็น โดยส่งอีเมลไปยังอาจารย์
2. เพิ่มการแสดงรายการผลงานวิจัยของผู้ใช้ โดยการนำเข้าข้อมูลจากเว็บไซต์ภายนอก

บรรณานุกรม

Bradbury, D. (2563, มกราคม 28). *Docker tutorial for beginners: How to get started with Docker*.

Retrieved from itpro.com:

<https://www.itpro.com/development/containers/354652/getting-started-with-docker>

Klebkomut, W. (2563, มกราคม 15). *Wiboon Klebkomut*. Retrieved from medium.com:

<https://medium.com/@wiboonklebkomut/อธิบาย-jquery-เบื้องต้น-3394b71eff28>

Ngamsai, H. (2562, มีนาคม 18). *SESSION ใน PHP คืออะไร?* Retrieved from medium.com:

<https://medium.com/@hattagornngamsai/session-ใน-php-คืออะไร-76d2573fce5d>

shareef.l. (2563, มกราคม 2). *PHP PDO*. Retrieved from oarkm.oas.psu.ac.th:

<https://oarkm.oas.psu.ac.th/index.php/blog/86>

Thanomwong, P. (2566, พฤศจิกายน 11). *การใช้งาน Bootstrap ในการสร้างหน้าเว็บ Responsive*. Retrieved

from borntodev.: <https://www.borntodev.com/2023/11/11/การใช้งาน-bootstrap-ในการสร้าง-หน้า/>

ภาคผนวก ก

การติดตั้งระบบ

ผู้ใช้งานสามารถดำเนินการติดตั้งระบบได้ตามขั้นตอนต่อไปนี้

1. ดาวน์โหลดและติดตั้งโปรแกรม Docker Desktop จาก
<https://www.docker.com/products/docker-desktop>
2. ทำการ Clone โครงการจาก <https://github.com/Akanitt/disboard>
3. แก้ไขชื่อไฟล์ .env.example สู่ .env และแก้ไขข้อมูลตัวแปรในไฟล์ .env สำหรับเครื่องคอมพิวเตอร์เครื่องผู้
 เครื่องผู้
4. แก้ไขข้อมูลในไฟล์ compose.yaml ดังต่อไปนี้
 - 4.1 พารามิเตอร์ environment ของ service server โดยมีค่าเดียวกับข้อมูลในไฟล์ .env
 - 4.2 พารามิเตอร์ environment ของ service db โดยมีค่าดังนี้
 - MARIADB_DATABASE: ชื่อของฐานข้อมูล
 - MARIADB_ROOT_PASSWORD: รหัสผ่านของผู้ใช้ในการเข้าถึงฐานข้อมูล
5. เปิดโปรแกรม terminal ในไดเรกทอรีโครงการ และรันคำสั่ง docker compose up --build

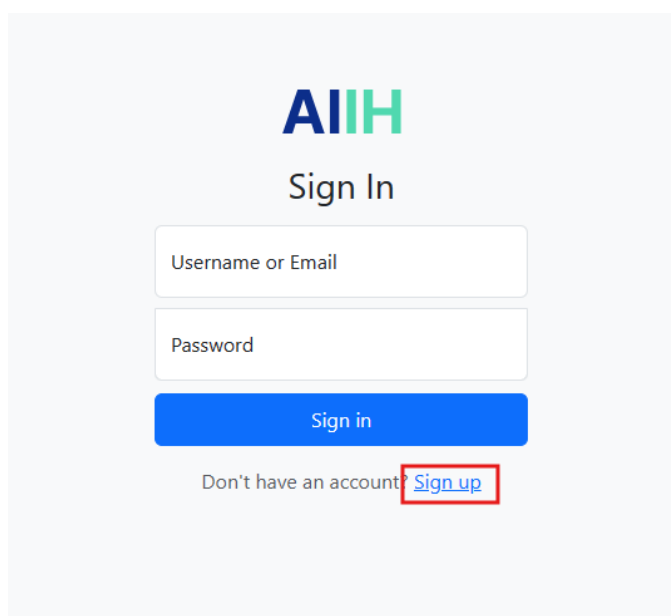
ภาคผนวก ข

วิธีการใช้งานระบบ

ผู้ใช้งานสามารถเข้าใช้งานระบบผ่านการเข้าถึง localhost:9000 เพื่อใช้งานแต่ละคุณสมบัติของเว็บไซต์ดังนี้

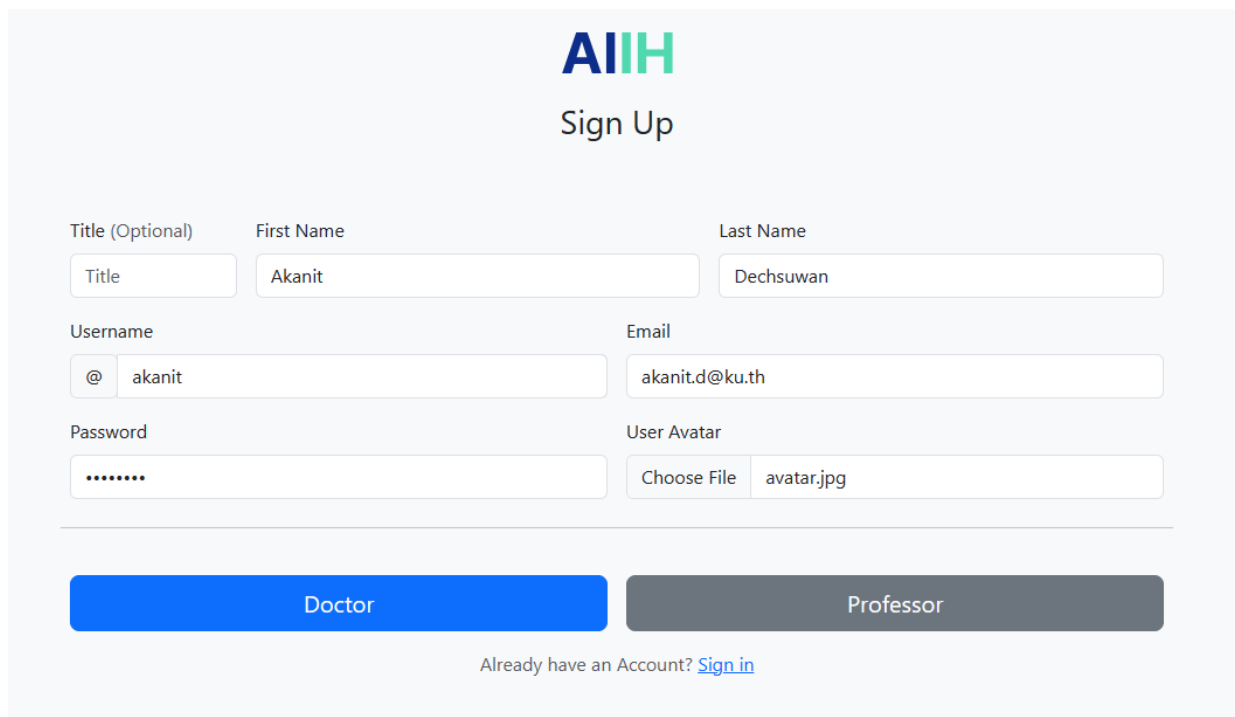
1. การสร้างบัญชีผู้ใช้

เมื่อผู้ใช้เรียกหน้าเว็บไซต์ จะปรากฏหน้าเข้าสู่ระบบ หากผู้ใช้ไม่มีบัญชีผู้ใช้ ให้ทำการเลือก Sign up บริเวณกรอบสีแดงของภาพ ข.1



ภาพที่ ข.1 หน้าการเข้าสู่ระบบและการชี้ไปยังหน้าการสมัครสมาชิก

ผู้ใช้กรอกข้อมูลลงในแบบฟอร์มการสมัครสมาชิกให้ถูกต้องและครบถ้วน โดยสามารถละเว้นคำนำหน้าและรูปภาพโปรไฟล์ได้ โดยระบบจะเรียกใช้ภาพเริ่มต้นแทน จากนั้นเลือกประเภทของผู้ใช้ หากเป็นแพทย์เลือก Doctor หรือหากเป็นอาจารย์เลือก Professor ดังภาพ ข.2



AIH
Sign Up

Title (Optional) First Name Last Name

Title Akanit Dechsuwan

Username Email

@ akanit akanit.d@ku.th

Password User Avatar

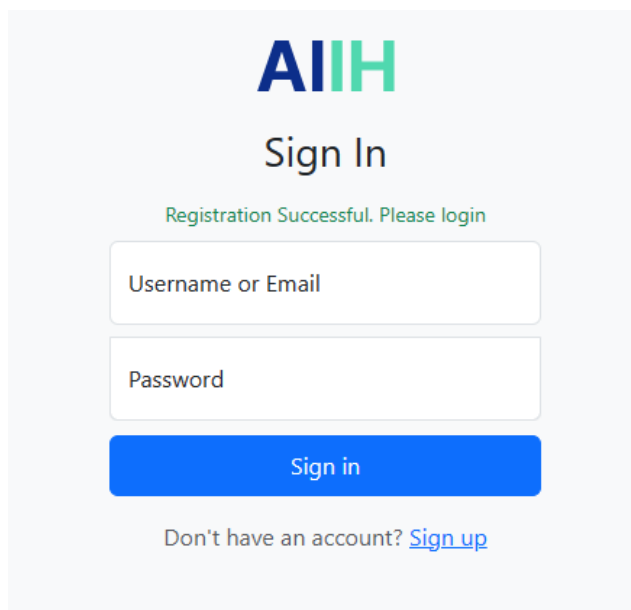
..... Choose File avatar.jpg

Doctor Professor

Already have an Account? [Sign in](#)

ภาพที่ ข.2 หน้าการสร้างบัญชีผู้ใช้พร้อมตัวอย่างข้อมูลในแบบฟอร์ม

เมื่อผู้ใ้กรอกข้อมูลถูกต้อง จะไปยังหน้าการเข้าสู่ระบบ พร้อมปรากฏข้อความแจ้งผู้้ว่าการสร้างบัญชีผู้ใช้สำเร็จ และสามารถใช้อ้ข้อมูลดังกล่าวเข้าสู่ระบบได้ ดังภาพ ข.3



AIH
Sign In

Registration Successful. Please login

Username or Email

Password

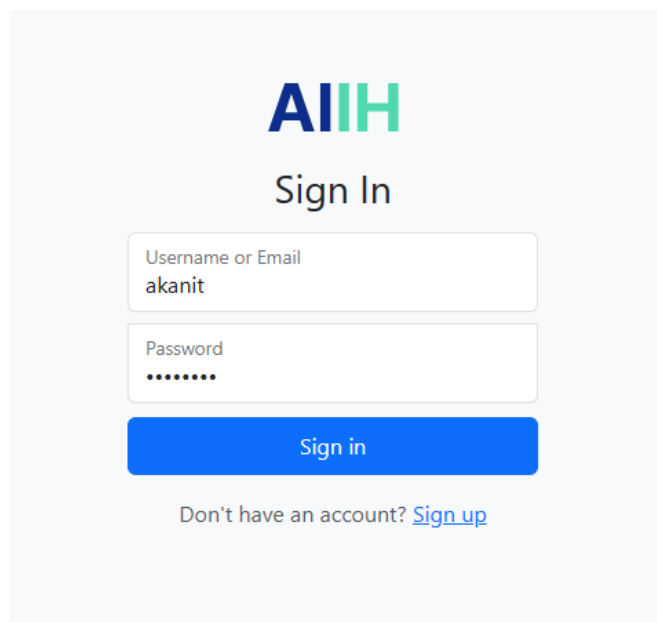
Sign in

Don't have an account? [Sign up](#)

ภาพที่ ข.3 หน้าการเข้าสู่ระบบและข้อความแสดงการสร้างบัญชีผู้ใช้สำเร็จ

2. การเข้าสู่ระบบ

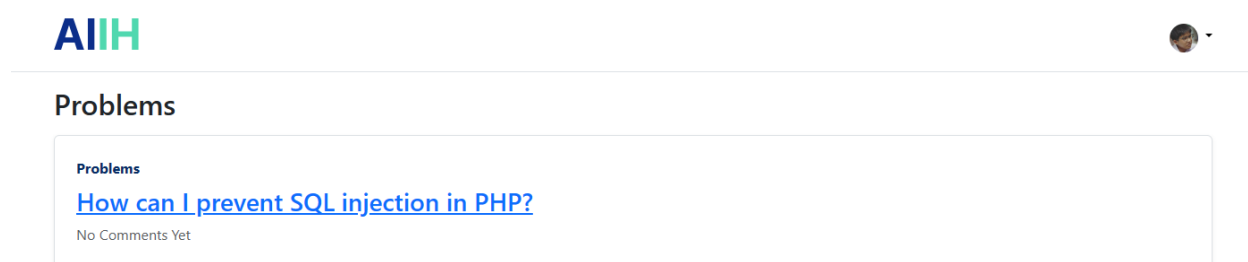
ผู้ใช้งานสามารถกรอกข้อมูลลงในแบบฟอร์มการเข้าสู่ระบบ ได้แก่ ชื่อบัญชีผู้ใช้หรืออีเมล และรหัสผ่าน จากนั้นเลือก Sign in เพื่อเข้าสู่ระบบ ดังภาพ ข.4



The image shows a 'Sign In' form for the AIH system. At the top is the AIH logo. Below it is the text 'Sign In'. There are two input fields: 'Username or Email' with the value 'akanit' and 'Password' with masked characters '.....'. A blue 'Sign in' button is below the fields. At the bottom, there is a link: 'Don't have an account? [Sign up](#)'.

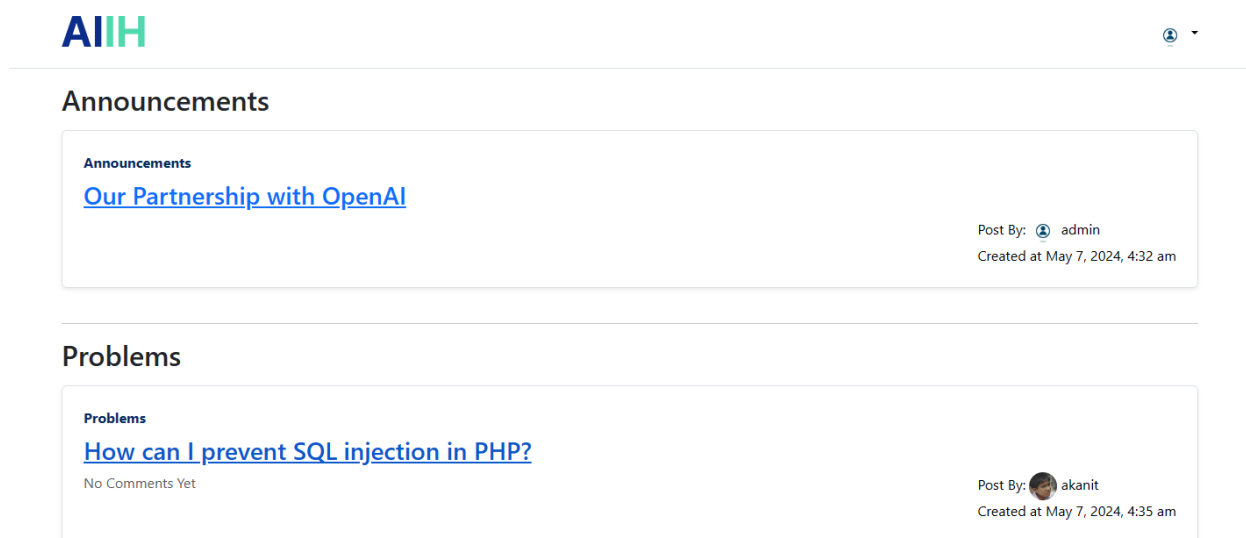
ภาพที่ ข.4 หน้าการสู่ระบบพร้อมตัวอย่างข้อมูล

หากผู้ใช้งานกรอกข้อมูลถูกต้องระบบจะไปยังหน้าแสดงโพสต์ทั้งหมดประเภทของผู้ใช้ หากผู้ใช้งานเป็นแพทย์จะแสดงเฉพาะโพสต์ประเภทปัญหาของตนเองเท่านั้น ดังภาพที่ ข.5



ภาพที่ ข.5 หน้าการแสดงโพสต์ทั้งหมดของแพทย์

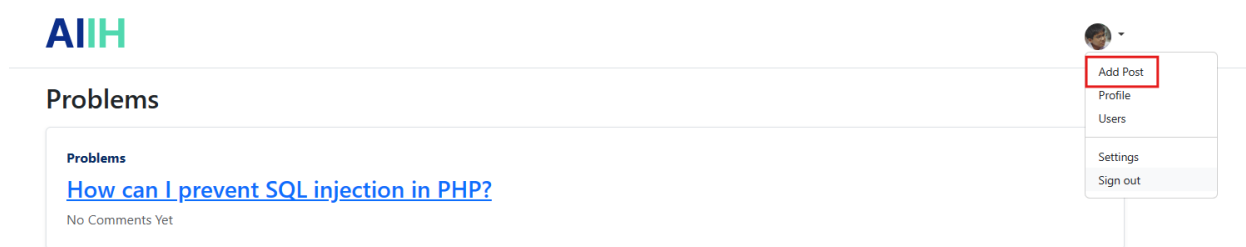
หากผู้ใช้เป็นอาจารย์จะแสดงโพสต์ทั้งหมด โดยแสดงโพสต์ประกาศ และโพสต์ปัญหา ดังภาพที่ ข.6



ภาพที่ ข.6 หน้าแสดงโพสต์ทั้งหมดของอาจารย์

3. การโพสต์

ผู้ใช้สามารถสร้างโพสต์ โดยการเลือกเครื่องหมายสามเหลี่ยมบริเวณบนขวา และเลือก Add Post ในบริเวณกรอบสีแดงของภาพ ข.7



ภาพที่ ข.7 หน้าแสดงโพสต์ทั้งหมดและการชี้ไปยังการสร้างโพสต์

หากผู้ใช้เป็นแพทย์ สามารถสร้างได้เพียงโพสต์ปัญหาเท่านั้น โดยกรอกหัวข้อเรื่องและรายละเอียด และสามารถเลือกเพิ่มเอกสารในโพสต์ได้ จากนั้นเลือก Post Problem ดังภาพ ข.8



Add Post

Title

Why shouldn't I use `mysql_*` functions in PHP?

Description

What are the technical reasons for why one shouldn't use `mysql_*` functions? (e.g. `mysql_query()`, `mysql_connect()` or `mysql_real_escape_string()`?)

Why should I use something else even if they work on my site?

If they don't work on my site, why do I get errors like

Warning: `mysql_connect()`: No such file or directory

Add File

Choose Files proposal.pdf

Post Problem

ภาพที่ ข.8 หน้าการสร้างโพสต์ของแพทย์พร้อมตัวอย่างข้อมูล

หากผู้ใช้เป็นอาจารย์ สามารถเลือกประเภทของโพสต์ได้โดยกรอกข้อมูลคล้ายในรูปแบบเดียวกับการโพสต์ของแพทย์ หากต้องการโพสต์ปัญหา เลือก Post Problem หรือต้องการโพสต์ประกาศเลือก Post Announcement ดังภาพ ข.9

Add Post

Title

Please don't make Careers developers-only in practice!

Description

Because careers.stackoverflow.com isn't careers.stackexchange.com (and uses some of the Stack Overflow design) most jobs advertised on Careers (which will become Stack Overflow) target developers.

Of course this issue predates the current change but it will enhance targeting for developers at the expense of sysadmins, thus making things far worse because there would be no longer separate site with a separate design and separate accounts (so targeting fewer people).

Going to Stack Overflow for finding non developers Jobs would be like the situation of this famous sketch.

Add File

Choose Files No file chosen

Post Problem

Post Announcements


ภาพที่ ข.9 หน้าการสร้างโพสต์ของอาจารย์พร้อมตัวอย่างข้อมูล

หากกรอกข้อมูลโพสต์ถูกต้อง ระบบจะไปยังหน้าโพสต์ทั้งหมดของผู้ใช้ และแสดงโพสต์ล่าสุดเป็นลำดับแรก
ดังภาพ ข.10

Announcements


Announcements

[Please don't make Careers developers-only in practice!](#)

Post By:  admin
Created at May 7, 2024, 5:09 am

Announcements

[Our Partnership with OpenAI](#)


Post By:  admin
Created at May 7, 2024, 4:32 am

Problems

Problems

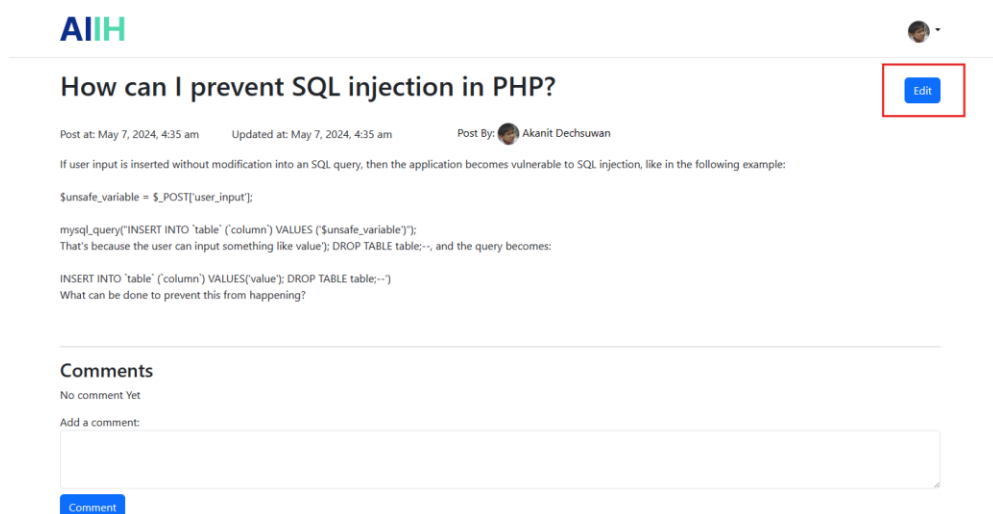
[How can I prevent SQL injection in PHP?](#)

No Comments Yet

Post By:  akanit
Created at May 7, 2024, 4:35 am

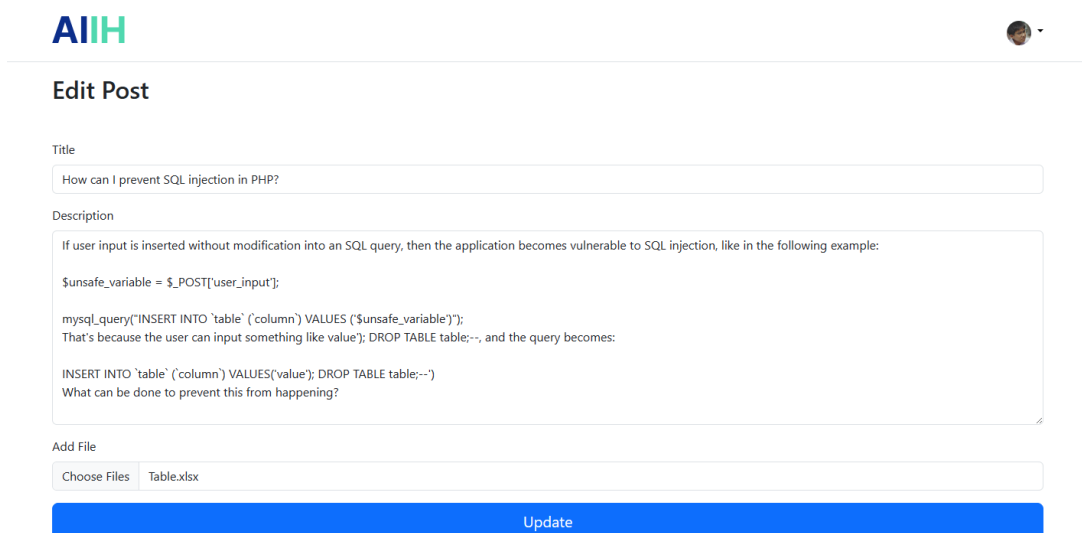
ภาพที่ ข.10 หน้าแสดงโพสต์ทั้งหมดของผู้ใช้ ซึ่งแสดงโพสต์ใหม่เป็นลำดับแรก

ผู้ใช้งานสามารถเข้าถึงและแก้ไขโพสต์ของตนเองได้ โดยเลือกโพสต์ที่ต้องการแก้ไข ระบบจะไปยังหน้าโพสต์ดังกล่าว และเลือก Edit ในบริเวณกรอบสีแดงของภาพ ข.11



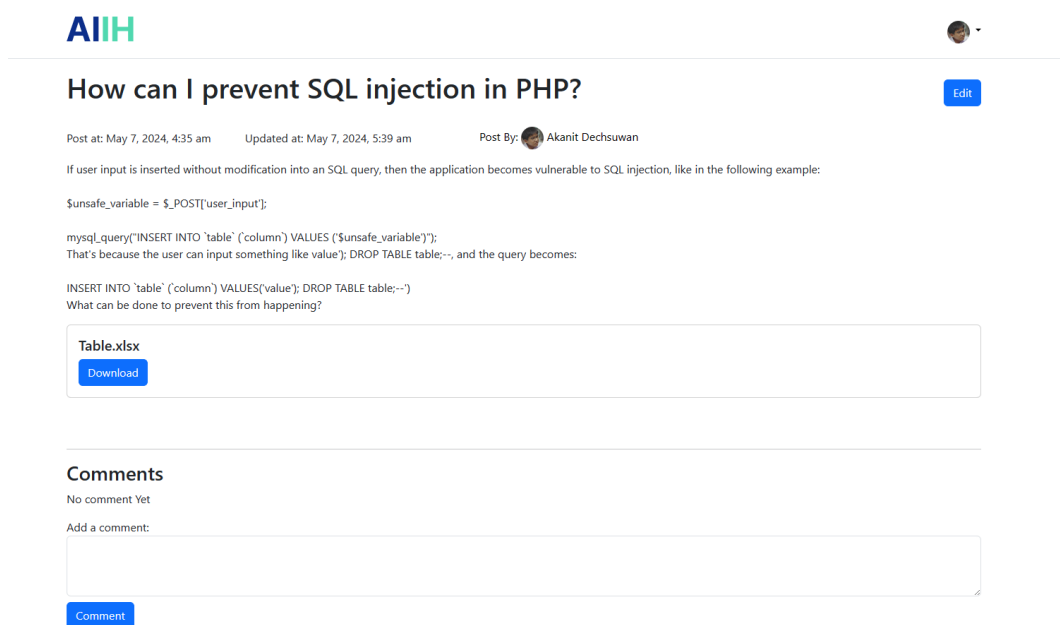
ภาพที่ ข.11 หน้าแสดงโพสต์และตัวเลือกในการแก้ไขโพสต์

ระบบจะไปยังหน้าการแก้ไขโพสต์ เมื่อผู้ใช้แก้ไขข้อมูลสำเร็จ เลือก Update ดังภาพ ข.12 ซึ่งทำการแก้ไขโพสต์โดยการเพิ่มเอกสารใหม่



ภาพที่ ข.12 หน้าการแก้ไขโพสต์โดยมีการเพิ่มเอกสาร

เมื่อแก้ไขโพสต์สำเร็จ ระบบจะไปยังหน้าโพสต์ที่ผ่านการแก้ไข พร้อมเปลี่ยนแปลงวันและเวลาในบริเวณ
Updated เป็นปัจจุบัน ดังภาพ ข.13



ภาพที่ ข.13 หน้าโพสต์ที่ผ่านการแก้ไขโดยแสดงเอกสารใหม่


4. การแสดงความคิดเห็น

อาจารย์สามารถแสดงความคิดเห็นบนโพสต์ของแพทย์ได้ โดยการเลือกโพสต์ที่ต้องการ จากนั้นแสดงความคิดเห็นลงในช่อง Add a comment ด้านล่างจากนั้น เลือก Comment ดังภาพ ข.14

How can I prevent SQL injection in PHP?

Post at: May 7, 2024, 4:35 am

Updated at: May 7, 2024, 5:39 am

Post By:  Akanit Dechsuwan

If user input is inserted without modification into an SQL query, then the application becomes vulnerable to SQL injection, like in the following example:

```
$unsafe_variable = $_POST['user_input'];
```

```
mysql_query("INSERT INTO `table` (`column`) VALUES ('$unsafe_variable');");
```

That's because the user can input something like value'; DROP TABLE table;--, and the query becomes:

```
INSERT INTO `table` (`column`) VALUES('value'); DROP TABLE table;--)
```

What can be done to prevent this from happening?

Table.xlsx

Download

Comments

No comment Yet

Add a comment:

The correct way to avoid SQL injection attacks, no matter which database you use, is to separate the data from SQL, so that data stays data and will never be interpreted as commands by the SQL parser.

Comment

ภาพที่ ข.14 หน้าโพสต์และการกรอกความคิดเห็นพร้อมตัวอย่างข้อมูล

เมื่อผู้ใช้กรอกความคิดเห็นสำเร็จ ระบบจะแสดงความคิดเห็นใหม่นั้นบนโพสต์ปัจจุบัน พร้อมระบุข้อมูลผู้แสดงความคิดเห็น และเปลี่ยนแปลงเวลาใน Updated เป็นปัจจุบัน ดังภาพ ข.15

How can I prevent SQL injection in PHP?

Post at: May 7, 2024, 4:35 am

Updated at: May 7, 2024, 5:50 am

Post By:  Akanit Dechsuwan

If user input is inserted without modification into an SQL query, then the application becomes vulnerable to SQL injection, like in the following example:

```
$unsafe_variable = $_POST['user_input'];
```

```
mysql_query("INSERT INTO `table` (`column`) VALUES ('$unsafe_variable')");
```

That's because the user can input something like value'); DROP TABLE table;--, and the query becomes:

```
INSERT INTO `table` (`column`) VALUES('value'); DROP TABLE table;--)
```

What can be done to prevent this from happening?

Table.xlsx

Download

Comments



Admin Disboard

May 7, 2024, 5:50 am

Professor

The correct way to avoid SQL injection attacks, no matter which database you use, is to separate the data from SQL, so that data stays data and will never be interpreted as commands by the SQL parser.

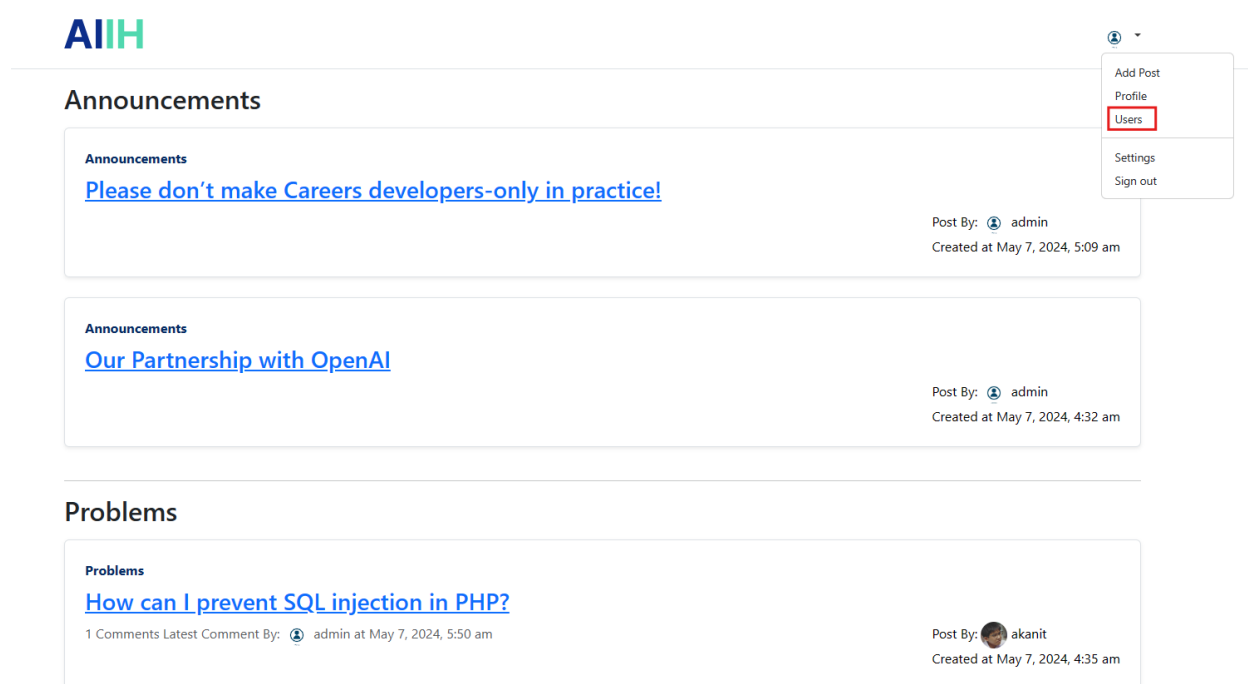
Add a comment:

Comment

ภาพที่ ข.15 หน้าโพสต์และความคิดเห็นใหม่ที่ใช้ทำการแสดงความคิดเห็น

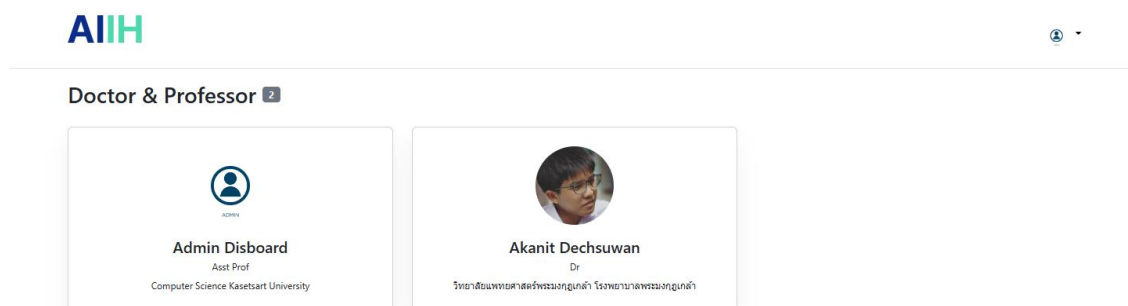
5. การเข้าถึงโปรไฟล์

ผู้ใช้งานสามารถเรียกดูโปรไฟล์ของผู้ใช้ทั้งหมดได้ โดยทำการเลือกเครื่องหมายสามเหลี่ยมบริเวณบนขวา และเลือก Users บริเวณกรอบสีแดงของภาพ ข.16



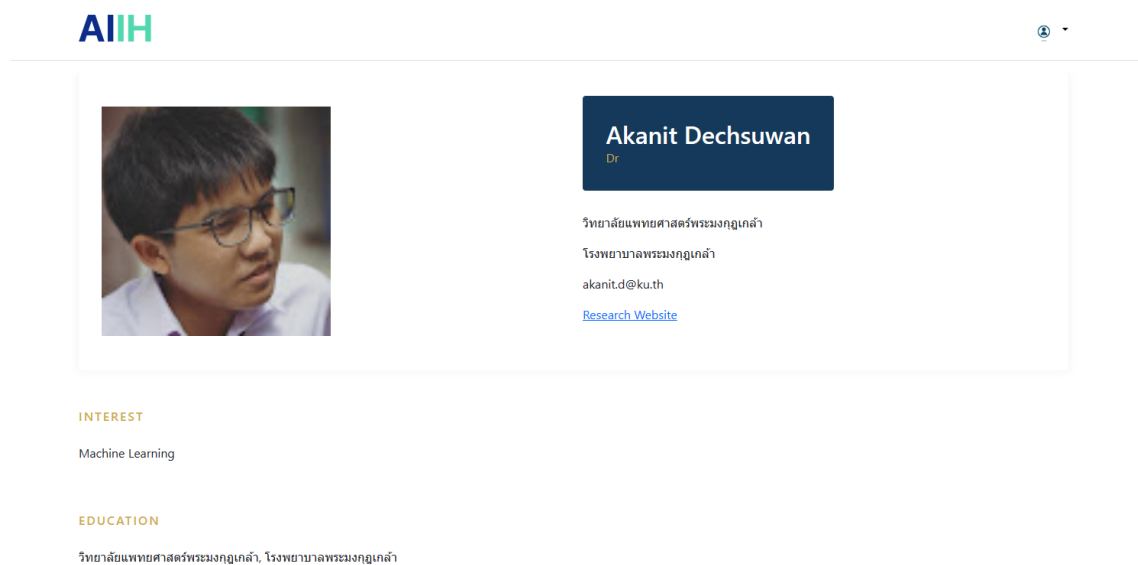
ภาพที่ ข.16 หน้าโพสต์ทั้งหมดและการชี้ไปยังหน้าแสดงข้อมูลผู้ใช้ทั้งหมด

ระบบแสดงข้อมูลผู้ใช้ทั้งหมดในลักษณะรายการซึ่งประกอบด้วยภาพโปรไฟล์ ชื่อและนามสกุล คำนำหน้าองค์กรและหน่วยงานที่สังกัด ดังภาพ ข.17



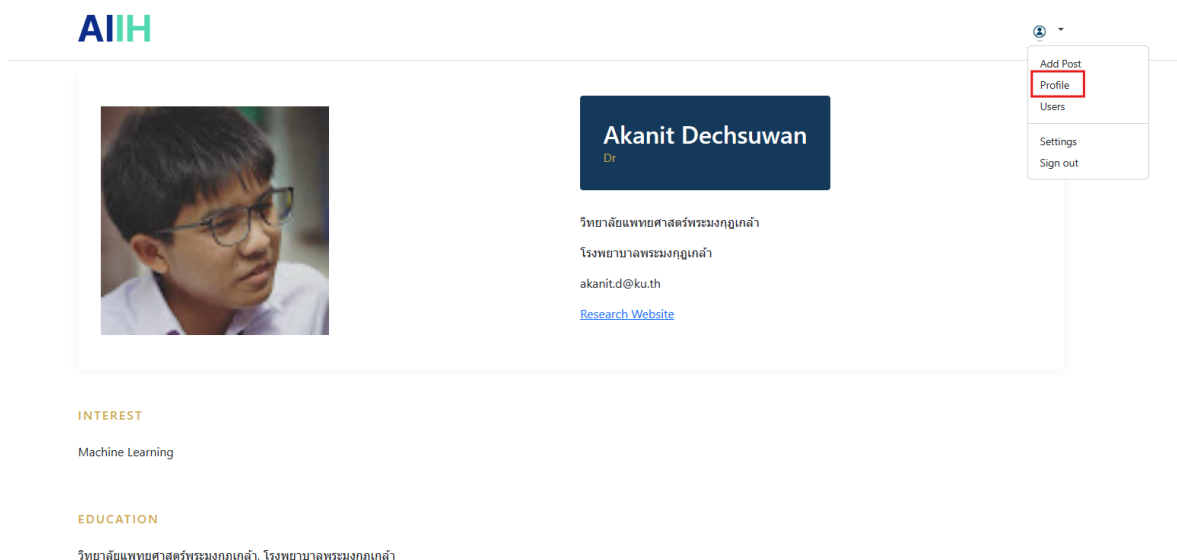
ภาพที่ ข.17 หน้าแสดงโปรไฟล์ของผู้ใช้ทั้งหมด

เมื่อต้องการเรียกดูโปรไฟล์ของผู้ใช้อื่น ทำการเลือกที่ข้อมูลของผู้ใช้ดังกล่าว ระบบจะไปยังหน้าผู้ใช้นั้น พร้อมแสดงข้อมูลความสนใจ ประวัติการศึกษา และเว็บไซต์ไปยังหน้าผลงานวิจัย ดังภาพ ข.18



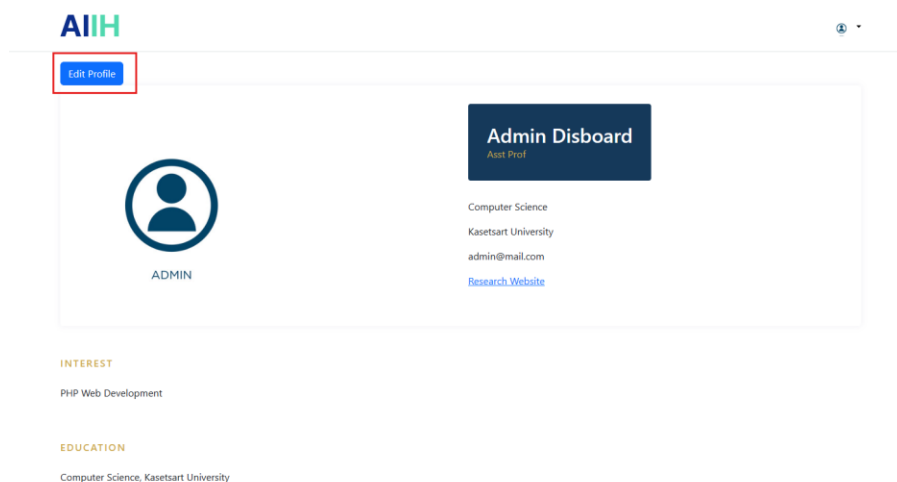
ภาพที่ ข.18 หน้าโปรไฟล์ของผู้ใช้รายบุคคล

หากผู้ใช้ต้องการเข้าถึงโปรไฟล์ของตนเอง สามารถเลือกโปรไฟล์ของตนเองในหน้าแสดงโปรไฟล์ทั้งหมด หรือเลือกเครื่องหมายสามเหลี่ยมบริเวณบนขวา และเลือก Profile ในบริเวณกรอบสีแดงของภาพ ข.19



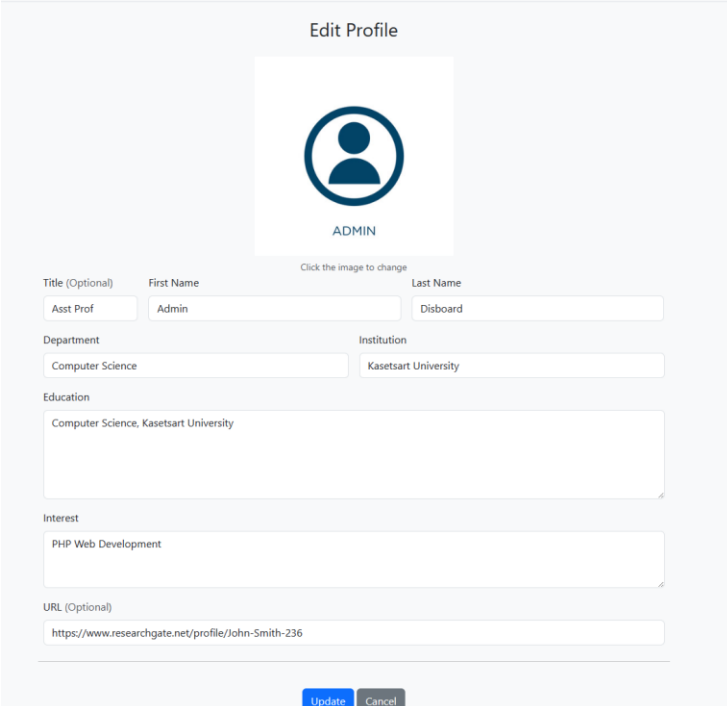
ภาพที่ ข.19 หน้าโปรไฟล์ของผู้อื่น และการชี้ไปยังหน้าโปรไฟล์ของตนเอง

ผู้ใช้สามารถแก้ไขข้อมูลในโปรไฟล์ของตนเองได้ โดยการเลือก Edit Profile ในหน้าโปรไฟล์ของตนเอง
บริเวณกรอบสีแดงของภาพ ข.20



ภาพที่ ข.20 แสดงหน้าโปรไฟล์ของผู้ใช้ และการชี้ไปยังหน้าการแก้ไขโปรไฟล์

ระบบจะดำเนินการแก้ไขโปรไฟล์ โดยเริ่มต้นผู้ใช้จะไม่มีข้อมูลแผนก หน่วยงาน ประวัติการศึกษา
ความสนใจ และลิงค์ไปยังเว็บไซต์ภายนอก เมื่อผู้ใช้แก้ไขข้อมูลสำเร็จเลือก Update ดังภาพ ข.21



Edit Profile

Click the image to change

ADMIN

Title (Optional) First Name Last Name

Asst Prof Admin Disboard

Department Institution

Computer Science Kasetsart University

Education

Computer Science, Kasetsart University

Interest

PHP Web Development

URL (Optional)

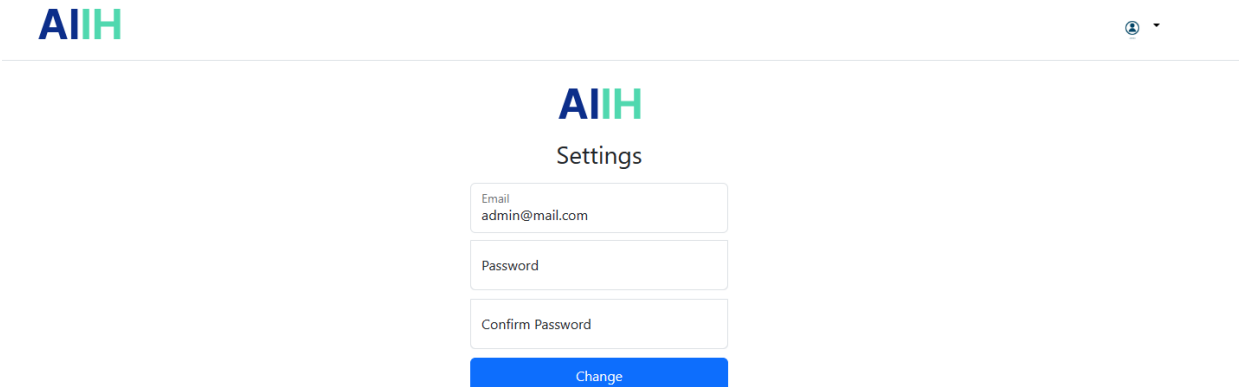
https://www.researchgate.net/profile/John-Smith-236

Update Cancel

ภาพที่ ข.21 แสดงหน้าการแก้ไขโปรไฟล์ของผู้ใช้พร้อมข้อมูลตัวอย่าง

6. การแก้ไขข้อมูลยืนยันตัวตนและการออกจากระบบ

ผู้ใช้สามารถแก้ไขอีเมล และรหัสผ่านของตนเองได้ โดยเลือกเครื่องหมายสามเหลี่ยมบริเวณบนขวา และเลือก Setting ระบบจะไปยังหน้าแก้ไขอีเมลและรหัสผ่าน ดังภาพ ข.22 เมื่อกรอกข้อมูลสำเร็จ ทำการเลือก Change จากนั้นระบบจะไปยังยังหน้าโพสต์ทั้งหมด หากการเปลี่ยนแปลงข้อมูลสำเร็จ



AIH

Settings

Email

admin@mail.com

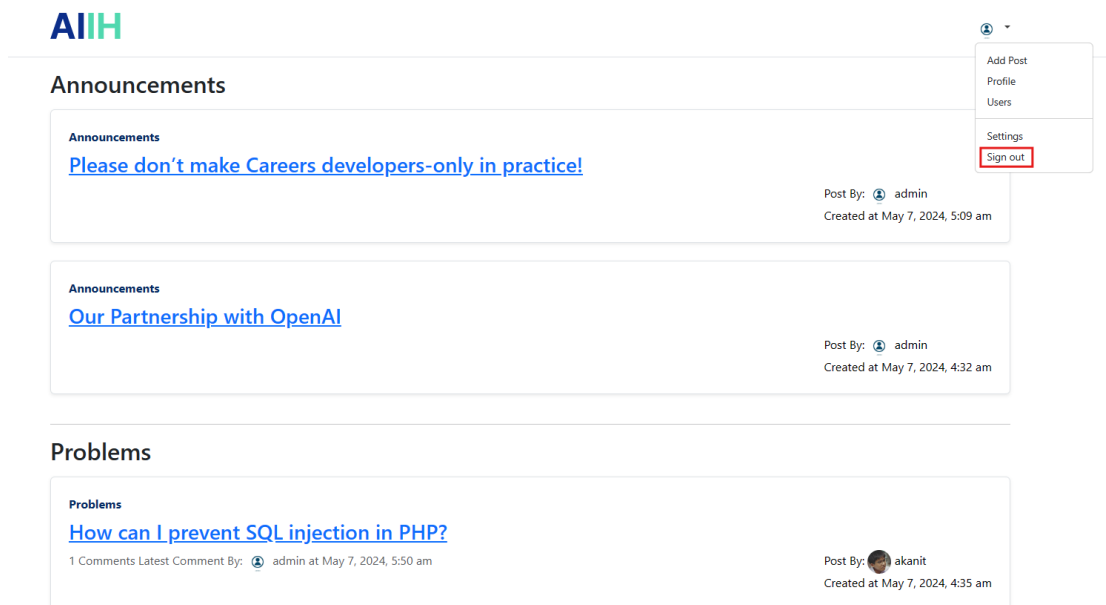
Password

Confirm Password

Change

ภาพที่ ข.22 หน้าการแก้ไขอีเมลและรหัสผ่าน

เมื่อต้องการสิ้นสุดการใช้งานผู้ใช้สามารถทำการออกจากระบบได้โดยการเลือกเครื่องหมายสามเหลี่ยมบริเวณบนขวา และเลือก Sign out บริเวณกรอบสีแดงของภาพ ข.23 ระบบจะไปยังหน้าการเข้าสู่ระบบอีกครั้ง



The screenshot shows the AIHH website interface. At the top left is the AIHH logo. Below it is the 'Announcements' section, which contains two posts. The first post is titled 'Please don't make Careers developers-only in practice!' and was posted by 'admin' on May 7, 2024, at 5:09 am. The second post is titled 'Our Partnership with OpenAI' and was also posted by 'admin' on May 7, 2024, at 4:32 am. Below the announcements is the 'Problems' section, which contains one post titled 'How can I prevent SQL injection in PHP?'. This post has 1 comment and was posted by 'akanit' on May 7, 2024, at 4:35 am. On the right side of the page, there is a user menu for 'admin' with options: 'Add Post', 'Profile', 'Users', 'Settings', and 'Sign out'. The 'Sign out' option is highlighted with a red rectangle.

ภาพที่ ข.23 หน้าโพสต์ทั้งหมดและการชี้ไปยังการออกจากระบบ