

โครงการเลขที่ วศ.คพ. P809-2/2566

เรื่อง

แพลตฟอร์มการแลกเปลี่ยนความคิดสร้างสรรค์ดิจิทัล: การเสริมสร้างการเรียนรู้ผ่าน
Gallery Walk

โดย

| | |
|--------------------------|----------------|
| นายญาณธิป ภู่อ่าง | รหัส 630612097 |
| นางสาวณัฐวรรณ เรียบเรียง | รหัส 630612099 |
| นายปิ่นฤทธิ์ร กันทรพย | รหัส 630612105 |

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ปีการศึกษา 2566

PROJECT No. CPE P809-2/2566

**Digital Creativity Exchange Platform: Enhancing Learning through
Gallery Walk**

Yanatip Bhoosawang 630612097

Nuttawan Reabreang 630612099

Panthon Kansap 630612105

**A Project Submitted in Partial Fulfillment of Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering
Faculty of Engineering
Chiang Mai University
2023**

หัวข้อโครงการ : แพลตฟอร์มการแลกเปลี่ยนความคิดสร้างสรรค์ดิจิทัล: การเสริมสร้างการเรียนรู้ผ่าน
Gallery Walk
: Digital Creativity Exchange Platform: Enhancing Learning through
Gallery Walk
โดย : นายญาณธิป ภู่อว่าง รหัส 630612097
นางสาวณัฐวรรณ เรียบเรียง รหัส 630612099
นายปิ่นธร กัณฑ์ทรัพย์ รหัส 630612105
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : ผศ. โดม โพธิ์กานนท์
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2566

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(ศ.ดร. สันติ พิทักษ์กัญญ์)

คณะกรรมการสอบโครงการ

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

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

..... กรรมการ
(อ.ดร. ธนาทิพย์ จันทรวง)

หัวข้อโครงการ : แพลตฟอร์มการแลกเปลี่ยนความคิดสร้างสรรค์ดิจิทัล: การเสริมสร้างการเรียนรู้ผ่าน
Gallery Walk
: Digital Creativity Exchange Platform: Enhancing Learning through
Gallery Walk

โดย : นายญาณธิป ภู่อ่าง รหัส 630612097
นางสาวณัฐวรรณ เรียบเรียง รหัส 630612099
นายปณัฏฐ์ กันทรทรัพย์ รหัส 630612105

ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : ผศ.โดม โพธิ์กานนท์
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2566

บทคัดย่อ

Gallery Walk คือการเรียนรู้แบบเดิมชมผลงาน ซึ่งเป็นเทคนิคการเรียนรู้ที่กระตุ้นให้ผู้เรียนแลกเปลี่ยนข้อมูล ความรู้หรือประสบการณ์ร่วมกันโดยใช้ผลงานที่ผู้เรียนแต่ละกลุ่มร่วมกันสร้างสรรค์ขึ้นเป็นสื่อกลางในการถ่ายทอดและใช้กระตือรือร้นสนทนาเป็นสื่อกลางในการรับคำติชมและข้อเสนอแนะจากผู้อื่น เพื่อนำข้อมูลป้อนกลับนั้นมาพัฒนาผลงานของกลุ่มหรือเติมเต็มความรู้ความเข้าใจของตัวผู้เรียนเอง

ในปัจจุบันการจัดกิจกรรมในรูปแบบ Gallery Walk นั้นมีความยุ่งยากในการจัดการ เนื่องจากผู้จัดการกิจกรรม (Event Manager) จะต้องจัดเตรียมสถานที่ สื่อการสื่อสาร และการจัดการเอกสารต่าง ๆ ที่เกี่ยวข้องกับกิจกรรม เป็นภาระต่อผู้จัดการกิจกรรม นอกจากนี้ยังไม่สามารถจัดกิจกรรมในรูปแบบ Hybrid ได้ ในกรณีที่ผู้เข้าร่วมงาน (Guest) ต้องการที่จะเยี่ยมชมผลงาน โครงการต่าง ๆ แบบออนไลน์ เพื่อให้ Virtual Money และแสดงความคิดเห็นต่าง ๆ และส่วนสุดท้ายไม่สามารถแสดงผลได้ทันทีว่าผลงานใดได้รับการประเมินสูงสุดหลังเสร็จสิ้นงานกิจกรรม

จากที่กล่าวมานั้น คณะผู้จัดทำจึงได้จัดทำโครงการเพื่อพัฒนาเว็บไซต์แอปพลิเคชันสำหรับการจัดกิจกรรม Gallery Walk ในรูปแบบ Hybrid ซึ่งสามารถให้ผู้จัดงานกิจกรรมสามารถจัดการกิจกรรมได้ง่ายขึ้น และสามารถให้ผู้เข้าร่วมงานสามารถเยี่ยมชมผลงาน โครงการต่าง ๆ แบบออนไลน์ และสามารถแสดงความคิดเห็นต่าง ๆ และส่วนสุดท้ายสามารถแสดงผลได้ทันทีว่าผลงานใดได้รับการประเมินสูงสุดหลังเสร็จสิ้นงานกิจกรรม

Project Title : Digital Creativity Exchange Platform: Enhancing Learning through Gallery Walk
Name : Yanatip Bhoosawang 630612097
Nuttawan Reabreang 630612099
Panthon Kansap 630612105
Department : Computer Engineering
Project Advisor : Asst.Prof.Dome Potikanond
Degree : Bachelor of Engineering
Program : Computer Engineering
Academic Year : 2023

ABSTRACT

Gallery Walk is the traditional way of learning and viewing works. This is a learning technique that encourages students to exchange information. Shared knowledge or experience by using the work that each group of learners jointly created as a medium for transferring and using a discussion board as a medium for receiving feedback and suggestions from others. To use that feedback to develop the group's work or complete the learners' own knowledge and understanding.

At present, organizing an activity in the form of a gallery walk is difficult to manage. This is because the event organizer (the event manager) must prepare the venue, communication media and managing various documents related to activities. It is a burden on event organizers. In addition, activities cannot be organized in a hybrid format in cases where attendees (guests) would like to visit the various projects online in order to provide virtual money and express various opinions. And the last part is that it cannot immediately show which work received the highest evaluation after the event has been completed.

From the above-mentioned The organizing team therefore created a project to develop a web application for organizing Gallery Walk activities in a hybrid format, which can allow event organizers to manage their activities more easily. And it can allow attendees to visit the various projects online and express various opinions, and the last part can immediately show which work received the highest evaluation after the event has finished.

กิตติกรรมประกาศ

โครงการนี้ได้เสร็จสมบูรณ์ด้วยความกรุณาอย่างยิ่งจาก ผศ.โดม โพธิกานนท์ ผู้ที่ได้มอบเวลาและความกรุณาเพื่อเป็นที่ปรึกษาในโครงการนี้ ได้ให้คำแนะนำที่มีคุณค่าและแนวทางที่ช่วยให้โครงการเกิดขึ้นอย่างสมบูรณ์และประสบความสำเร็จ

ไม่เพียงแต่นั้นเท่านั้น ขอขอบคุณอย่างสูงสำหรับคำปรึกษาจาก อ.ดร.ชินวัตร อิศราดิศัยกุล และ ผศ.ดร.ธนาทิพย์ จันทรงค์ ที่ได้มอบคำแนะนำและเสนอแนวทางที่มีประสิทธิภาพ ทำให้โครงการเกิดประสิทธิภาพมากขึ้น

นอกจากนี้ เราต้องขอบคุณเพื่อนร่วมทีมทุกคนที่ให้การสนับสนุนและกำลังใจ และขอบคุณครอบครัวที่เป็นกำลังใจสำคัญในการผ่านช่วงเวลาที่ทำหายไปด้วยกัน ขอขอบคุณทุกท่านที่เคยให้คำแนะนำและความช่วยเหลือ โดยที่ไม่ระบุนาม ทุกคำปรึกษาและการสนับสนุนที่มีค่ามากสำหรับเรา

หากหนังสือโครงการเล่มนี้มีข้อผิดพลาดประการใดผู้จัดทำขอน้อมรับด้วย ความยินดียิ่งและขออภัยมา ณ ที่นี้ด้วย

นายญาณธิป ภู่อ่าง
นางสาวณัฐวรรณ เรียบเรียง
นายปณัฏฐ์ร กัณฑ์พร
20 มีนาคม 2567

สารบัญ

| | |
|---|-----------|
| บทคัดย่อ | ข |
| Abstract | ค |
| กิตติกรรมประกาศ | ง |
| สารบัญ | จ |
| สารบัญรูป | ช |
| สารบัญตาราง | ซ |
| 1 บทนำ | 1 |
| 1.1 ที่มาของโครงการ | 1 |
| 1.2 วัตถุประสงค์ของโครงการ | 1 |
| 1.3 ขอบเขตของโครงการ | 1 |
| 1.3.1 ขอบเขตด้านฮาร์ดแวร์ | 1 |
| 1.3.2 ขอบเขตด้านซอฟต์แวร์ | 1 |
| 1.4 ประโยชน์ที่ได้รับ | 2 |
| 1.5 เทคโนโลยีและเครื่องมือที่ใช้ | 2 |
| 1.5.1 เทคโนโลยีด้านฮาร์ดแวร์ | 2 |
| 1.5.2 เทคโนโลยีด้านซอฟต์แวร์ | 2 |
| 1.6 แผนการดำเนินงาน | 3 |
| 1.7 บทบาทและความรับผิดชอบ | 3 |
| 2 ทฤษฎีที่เกี่ยวข้อง | 4 |
| 2.1 ด้านโครงสร้างเว็บแอปพลิเคชัน | 4 |
| 2.1.1 MVC Architecture | 4 |
| 2.1.2 RESTful API | 4 |
| 2.1.3 ระบบฐานข้อมูล (Database System) | 5 |
| 2.2 ด้านเทคโนโลยี | 6 |
| 2.2.1 HTML | 6 |
| 2.2.2 CSS | 7 |
| 2.2.3 TypeScript | 7 |
| 2.2.4 React JS | 7 |
| 2.2.5 Mantine UI | 8 |
| 2.2.6 PostgreSQL | 8 |
| 2.2.7 Prisma | 9 |
| 2.3 ด้าน User Interface | 10 |
| 2.3.1 Design Thinking | 10 |
| 2.4 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ | 11 |
| 3 โครงสร้างและขั้นตอนการทำงาน | 13 |
| 3.1 หลักการทำงานของระบบ | 13 |
| 3.1.1 การทำงานของระบบ (System Architecture) | 13 |
| 3.1.2 เส้นทางของผู้ใช้ (User Flow) | 14 |
| 3.1.3 โครงสร้างฐานข้อมูล (Database Schema) | 15 |
| 3.2 ส่วนเชื่อมต่อระหว่างผู้ใช้งานกับระบบ (User Interface) | 15 |
| 3.2.1 ผู้ใช้งาน (User) | 15 |
| 3.2.2 ผู้ดูแลระบบ (Admin) | 20 |

| | | |
|----------|--|-----------|
| 4 | การทดลองและผลลัพธ์ | 27 |
| 4.1 | การทดลองเกี่ยวกับการทำงานของระบบ | 27 |
| 4.1.1 | ผู้ใช้ทั่วไป | 27 |
| 4.1.2 | ทันตแพทย์ | 27 |
| 4.1.3 | ทันตบุคลากร | 28 |
| 4.1.4 | อาสาสมัครสาธารณสุขประจำหมู่บ้าน (อสม.) | 28 |
| 5 | บทสรุปและข้อเสนอแนะ | 29 |
| 5.1 | สรุปผล | 29 |
| 5.2 | ปัญหาที่พบและแนวทางการแก้ไข | 29 |
| 5.3 | ข้อเสนอแนะและแนวทางการพัฒนาต่อ | 29 |
| | บรรณานุกรม | 30 |
| ก | The first appendix | 32 |
| ก.1 | Appendix section | 32 |
| ข | คู่มือการใช้งานระบบ | 33 |
| | ประวัติผู้เขียน | 34 |

សារប័ណ្ណ

| | | |
|------|------------|----|
| 2.1 | HTML | 7 |
| 2.2 | React JS | 8 |
| 2.3 | Mantine UI | 8 |
| 2.4 | PostgreSQL | 9 |
| 2.5 | Prisma ORM | 10 |
| 2.6 | Poem | 11 |
| 3.1 | Poem | 13 |
| 3.2 | Poem | 14 |
| 3.3 | Poem | 15 |
| 3.4 | Poem | 16 |
| 3.5 | Poem | 17 |
| 3.6 | Poem | 18 |
| 3.7 | Poem | 19 |
| 3.8 | Poem | 20 |
| 3.9 | Poem | 21 |
| 3.10 | Poem | 22 |
| 3.11 | Poem | 23 |
| 3.12 | Poem | 23 |
| 3.13 | Poem | 24 |
| 3.14 | Poem | 24 |
| 3.15 | Poem | 25 |
| 3.16 | Poem | 25 |
| 3.17 | Poem | 26 |

สารบัญตาราง

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

โครงการ Gallery Walk มีความหมายในการให้ผู้เรียนมีโอกาสดูแลเปลี่ยนความรู้และประสบการณ์ผ่านผลงานที่สร้างขึ้นเป็นสื่อกลาง และการใช้กระดานสนทนาเพื่อรับคำแนะนำและคำชมเชยจากผู้อื่น เพื่อให้สามารถพัฒนาผลงานหรือเพิ่มเติมความรู้แก่ตนเองได้

การจัดกิจกรรมแบบ Gallery Walk ในปัจจุบันมีความยุ่งยากในการจัดการ เนื่องจากต้องมีการจัดเตรียมสถานที่ สื่อการสื่อสาร และการจัดการเอกสารต่าง ๆ ซึ่งเป็นภาระต่อผู้จัดงาน นอกจากนี้ยังไม่สามารถจัดกิจกรรมแบบ Hybrid ได้ โดยเฉพาะในกรณีที่ผู้เข้าร่วมต้องการที่จะเยี่ยมชมผลงานแบบออนไลน์และให้คำแนะนำ

ดังนั้น เพื่อแก้ไขปัญหาดังกล่าว ทีมผู้จัดงานได้พัฒนาเว็บแอปพลิเคชันสำหรับการจัดกิจกรรม Gallery Walk ในรูปแบบ Hybrid เพื่อให้การจัดการกิจกรรมเป็นไปอย่างมีประสิทธิภาพ และเพื่อให้ผู้เข้าร่วมสามารถเยี่ยมชมผลงานและให้ความเห็นออนไลน์ได้ นอกจากนี้ยังสามารถแสดงผลการประเมินได้ทันทีหลังจากเสร็จสิ้นกิจกรรม

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อให้การจัดการกิจกรรมของผู้จัดกิจกรรมสะดวกและมีประสิทธิภาพมากขึ้น
2. เพื่อสร้างช่องทางสำหรับการดำเนินกิจกรรมทางออนไลน์ให้กับผู้เข้าร่วมกิจกรรม สามารถจัดกิจกรรมในรูปแบบ Hybrid ได้
3. การประเมินที่เน้นการมีส่วนร่วมและรับความคิดเห็น ซึ่งส่งเสริมการพัฒนาผลงานของผู้นำเสนอโครงการ

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตด้านฮาร์ดแวร์

โครงการนี้ต้องการฮาร์ดแวร์ต่อไปนี้ จึงจะสามารถใช้งานได้มีประสิทธิภาพ

- คอมพิวเตอร์ส่วนบุคคลหรือโทรศัพท์มือถือที่สามารถใช้งานเว็บเบราว์เซอร์ได้

1.3.2 ขอบเขตด้านซอฟต์แวร์

โครงการนี้ต้องการซอฟต์แวร์ต่อไปนี้ จึงจะสามารถใช้งานได้มีประสิทธิภาพ

- สามารถใช้งานเว็บไซต์บนระบบปฏิบัติการทั่วไปได้ เช่น Windows, macOS, Linux, Android, iOS และอื่น ๆ

โดยการแสดงผลของเว็บไซต์จะแสดงผลได้เหมาะสม ตามอุปกรณ์ที่ใช้งานและตามบทบาทผู้ใช้งานดังต่อไปนี้

- ผู้จัดการกิจกรรม (Event Manager): คอมพิวเตอร์ส่วนบุคคล
- ผู้นำเสนอโครงการ (Presenter): คอมพิวเตอร์ส่วนบุคคล
- แขกผู้เข้าร่วมกิจกรรม (Guest): คอมพิวเตอร์ส่วนบุคคลหรือโทรศัพท์มือถือ

1.4 ประโยชน์ที่ได้รับ

ผู้ใช้งาน

- ผู้จัดการกิจกรรม (Event Manager)
 - สามารถจัดการกิจกรรมได้อย่างมีประสิทธิภาพ
 - สามารถจัดการกิจกรรมแบบ Hybrid ได้
 - สามารถแสดงผลการประเมินได้ทันทีหลังจากเสร็จสิ้นกิจกรรม
- ผู้นำเสนอโครงการ (Presenter)
 - สามารถนำเสนอผลงานของตนได้อย่างมีประสิทธิภาพ
 - สามารถรับความคิดเห็นและคำแนะนำจากผู้เข้าร่วมกิจกรรมได้อย่างมีประสิทธิภาพ
- ผู้เข้าร่วมกิจกรรม (Guest)
 - สามารถเยี่ยมชมผลงานและให้ความเห็นออนไลน์ได้อย่างมีประสิทธิภาพ

ผู้พัฒนา

- ได้รับความรู้และความเข้าใจในการพัฒนาเว็บแอปพลิเคชัน
- ได้ฝึกทักษะในการใช้เทคโนโลยีที่เกี่ยวข้องกับการพัฒนาเว็บแอปพลิเคชัน
- ได้ฝึกทักษะในการทำงานเป็นทีมและทักษะในการวิเคราะห์และแก้ไขปัญหาที่อาจเกิดขึ้นในการพัฒนา

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านฮาร์ดแวร์

1.5.2 เทคโนโลยีด้านซอฟต์แวร์

- ภาษาโปรแกรมมิ่ง: TypeScript, JavaScript, HTML, CSS
- ฐานข้อมูล: PostgreSQL, Prisma ORM, Minio (Object Storage)
- เครื่องมือและเทคโนโลยี: React, Mantine UI, Git, GitHub

1.6 แผนการดำเนินงาน

| ขั้นตอนการดำเนินงาน | ก.ค. 2566 | ส.ค. 2566 | ก.ย. 2566 | ต.ค. 2566 | พ.ย. 2566 | ธ.ค. 2566 | ม.ค. 2567 | ก.พ. 2567 | มี.ค. 2567 |
|---|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| คิดหัวข้อและนำเสนอหัวข้อโครงการ | | | | | | | | | |
| ศึกษาค้นคว้าเกี่ยวกับเทคโนโลยีที่เกี่ยวข้อง | | | | | | | | | |
| ออกแบบ UI/UX ของเว็บแอปพลิเคชัน | | | | | | | | | |
| พัฒนาเว็บแอปพลิเคชัน | | | | | | | | | |
| ทดสอบและปรับปรุงเว็บแอปพลิเคชัน | | | | | | | | | |

1.7 บทบาทและความรับผิดชอบ

มีหน้าที่และความรับผิดชอบ ดังนี้

นายญาณธิป ภู่อ่าง รหัสนักศึกษา 630612097 รับผิดชอบในการศึกษาค้นคว้าเทคโนโลยีที่เกี่ยวข้อง, ออกแบบโครงสร้างของเว็บแอปพลิเคชันและพัฒนาเว็บแอปพลิเคชันส่วนหน้าบ้าน (Frontend)

นางสาวณัฐวรรณ เรียบเรียง รหัสนักศึกษา 630612099 รับผิดชอบในการศึกษาค้นคว้าเทคโนโลยีที่เกี่ยวข้อง, ออกแบบโครงสร้างของเว็บแอปพลิเคชันและพัฒนาเว็บแอปพลิเคชันส่วนหน้าบ้าน (Frontend)

นายปณัฏฐ์ กันทรัพย์ รหัสนักศึกษา 630612105 รับผิดชอบในการศึกษาค้นคว้าเทคโนโลยีที่เกี่ยวข้อง, ออกแบบโครงสร้างของเว็บแอปพลิเคชันและพัฒนาเว็บแอปพลิเคชันส่วนหลังบ้าน (Backend)

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ด้านโครงสร้างเว็บแอปพลิเคชัน

ในส่วนนี้จะอธิบายถึงโครงสร้างของเว็บแอปพลิเคชันที่ใช้ในการพัฒนา

2.1.1 MVC Architecture

MVC [1] เป็นตัวย่อของคำว่า Model View Controller ใช้เรียกรูปแบบการพัฒนาซอฟต์แวร์ที่มีโครงสร้างซึ่งแบ่งออกเป็น 3 ส่วนหลัก ตามตัวย่อของชื่อ รูปแบบการพัฒนาซอฟต์แวร์แบบ MVC ถูกนำไปใช้ในขั้นตอนการพัฒนาหลากหลายภาษา เพราะ MVC เป็นเพียงหลักการออกแบบโปรแกรม (Design Pattern) รูปแบบหนึ่งเท่านั้น ซึ่งเป็นที่นิยมมาก ในการนำมาพัฒนาแอปพลิเคชันซอฟต์แวร์แต่ละแพลตฟอร์มและประยุกต์ใช้ในอีกหลาย ๆ ด้าน

ส่วนของ Model (M)

model คือส่วนของการเก็บรวบรวมข้อมูล ไม่ว่าข้อมูลนั้น ๆ จะถูกจัดเก็บในรูปแบบใดก็ตาม ในฐานข้อมูลแบบเป็น Object Class หรือที่นิยมเรียกกันว่า VO (Value Object) หรือเก็บเป็นไฟล์ข้อมูลเลย เมื่อข้อมูลถูกโหลดเข้ามาจากที่ต่าง ๆ และเข้ามายังส่วนของโมเดล ตัวโมเดลจะทำการจัดการเตรียมข้อมูลให้เป็นรูปแบบที่เหมาะสม เพื่อรอการร้องขอข้อมูลจากส่วนของ Controller

ส่วนของ View (V)

view คือส่วนของการแสดงผล หรือส่วนที่จะปฏิสัมพันธ์กับผู้ใช้งาน (User Interface) หน้าที่ของ view ในการเขียนโปรแกรมแบบ MVC คือคอยรับคำสั่งจากส่วนของ Controller และ End User เริ่มแรกเลยตัว view อาจจะรับคำสั่งจาก Controller ให้แสดงผลหน้า Home และเมื่อผู้ใช้งานหน้าเว็บกดปุ่มสั่งซื้อ View จะส่งข้อมูลไปให้ Controller เพื่อประมวลผลและแสดงบางอย่างจาก Action นั้น

ส่วนของ Controller (C)

controller คือส่วนของการเริ่มทำงาน และรับคำสั่ง โดยที่คำสั่งนั้นจะเกิดขึ้นในส่วนการติดต่อกับผู้ใช้งานคือ view เมื่อผู้ใช้งานทำการ Interactive กับ UI view จะเกิดเหตุการณ์หรือข้อมูลบางอย่างขึ้น ตัว view จะส่งข้อมูลนั้น มายัง controller ตัว controller จะทำการประมวลผลโดยบางคำสั่งอาจจะต้องไปติดต่อกับ model ก่อน เพื่อทำการประมวลผลข้อมูลอย่างถูกต้องเรียบร้อยแล้วก็จะส่งไปยัง view เพื่อแสดงผลตามคำสั่งที่ end user ร้องขอมา Controller จะทำหน้าที่เป็นตัวกลางระหว่าง Model และ View ให้ทำงานร่วมกันอย่างมีประสิทธิภาพและตรงกับ ความต้องการของ End User มากที่สุด

2.1.2 RESTful API

RESTful API [2] เป็นอินเทอร์เฟซที่ระบบคอมพิวเตอร์สองระบบใช้เพื่อแลกเปลี่ยนข้อมูลผ่านอินเทอร์เน็ตได้อย่างปลอดภัย แอปพลิเคชันทางธุรกิจส่วนใหญ่ต้องสื่อสารกับแอปพลิเคชันภายในอื่นๆ และของบุคคล

ที่สามเพื่อทำงานต่างๆ ตัวอย่างเช่น หากต้องการสร้างสลิปเงินเดือน ระบบบัญชีภายในของคุณต้องแบ่งปันข้อมูลกับระบบธนาคารของลูกค้าเพื่อออกใบแจ้งหนี้และสื่อสารกับแอปพลิเคชันบันทึกเวลาปฏิบัติงานภายใน โดยอัตโนมัติ RESTful API ให้การสนับสนุนการแลกเปลี่ยนข้อมูลนี้เพราะเป็นระบบที่มีมาตรฐานการสื่อสารระหว่างซอฟต์แวร์ที่ปลอดภัย เสถียร และมีประสิทธิภาพ

API (Application Programming Interface)

ส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface หรือ API) กำหนดกฎที่คุณต้องปฏิบัติตามเพื่อสื่อสารกับระบบซอฟต์แวร์อื่น โดยนักพัฒนาเปิดเผยหรือสร้าง API เพื่อให้แอปพลิเคชันอื่นสามารถสื่อสารกับแอปพลิเคชันของตนได้ทางโปรแกรม ตัวอย่างเช่น แอปพลิเคชันบันทึกเวลาปฏิบัติงานแสดง API ที่ขอชื่อเต็มของพนักงานและช่วงวันที่ เมื่อได้รับข้อมูลนี้แล้ว ระบบจะประมวลผลบันทึกเวลาปฏิบัติงานของพนักงานเป็นการภายใน และส่งกลับจำนวนชั่วโมงที่ทำงานในช่วงวันที่ดังกล่าว ทั้งนี้คุณสามารถมองได้ว่า API เว็บเป็นเกตเวย์ระหว่างไคลเอนต์และทรัพยากรบนเว็บ

ไคลเอนต์ ไคลเอนต์คือผู้ใช้ที่ต้องการเข้าถึงข้อมูลจากเว็บ โดยไคลเอนต์อาจเป็นบุคคลหรือระบบซอฟต์แวร์ที่ใช้ API ก็ได้ ตัวอย่างเช่น นักพัฒนาสามารถเขียนโปรแกรมที่เข้าถึงข้อมูลสภาพอากาศจากระบบสภาพอากาศ หรือคุณสามารถเข้าถึงข้อมูลเดียวกันจากเบราว์เซอร์เมื่อคุณเยี่ยมชมเว็บไซต์รายงานสภาพอากาศได้โดยตรง

ทรัพยากร ทรัพยากรคือข้อมูลที่แอปพลิเคชันต่างๆ มอบให้แก่ไคลเอนต์ โดยทรัพยากรอาจเป็นรูปภาพ วิดีโอ ข้อความ ตัวเลข หรือข้อมูลประเภทใดก็ได้ ทั้งนี้เครื่องคอมพิวเตอร์ที่มอบทรัพยากรให้แก่ไคลเอนต์นั้นเรียกอีกอย่างว่าเซิร์ฟเวอร์ องค์กรต่างๆ ใช้ API เพื่อแบ่งปันทรัพยากรและให้บริการเว็บในขณะที่ยังคงดูแลรักษาความปลอดภัย การควบคุม และการรับรองความถูกต้องไปพร้อมกัน นอกจากนี้ API ยังช่วยให้ลูกค้าระบุได้ว่าไคลเอนต์ใดสามารถเข้าถึงทรัพยากรภายในที่เฉพาะเจาะจงได้

REST (Representational State Transfer)

REST เป็นสถาปัตยกรรมซอฟต์แวร์ที่กำหนดเงื่อนไขว่า API ควรทำงานอย่างไร โดยแต่แรกเริ่มนั้น มีการสร้าง REST ขึ้นเพื่อเป็นแนวทางในการจัดการการสื่อสารบนเครือข่ายที่ซับซ้อน เช่น อินเทอร์เน็ต คุณสามารถใช้สถาปัตยกรรม REST เพื่อรองรับการสื่อสารที่มีประสิทธิภาพสูงและเชื่อถือได้ในทุกระดับ คุณยังสามารถใช้และปรับเปลี่ยนสถาปัตยกรรมได้อย่างง่ายดาย โดยนำความสามารถในการมองเห็นและการเคลื่อนย้ายข้ามแพลตฟอร์มมาสู่ทุกระบบ API

นักพัฒนา API สามารถออกแบบ API ได้โดยใช้สถาปัตยกรรมต่างๆ โดย API ที่เป็นไปตามรูปแบบสถาปัตยกรรม REST เรียกว่า REST API บริการเว็บที่ใช้สถาปัตยกรรม REST เรียกว่าบริการเว็บ RESTful คำว่า RESTful API โดยทั่วไปหมายถึง API เว็บแบบ RESTful อย่างไรก็ตาม คุณสามารถใช้คำว่า REST API และ RESTful API แทนกันได้

2.1.3 ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล (Database System) [3] คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบ มีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูลเกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งาน และดูแล

รักษาป้องกันข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่าง ผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (data base management system) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

ประโยชน์ของฐานข้อมูล

1. ลดการเก็บข้อมูลที่ซ้ำซ้อน

ข้อมูลบางชุดที่อยู่ในรูปของแฟ้มข้อมูลอาจมีปรากฏอยู่หลาย ๆ แห่ง เพราะมีผู้ใช้ข้อมูลชุดนี้หลายคน เมื่อใช้ระบบฐานข้อมูลแล้วจะช่วยให้ความซ้ำซ้อนของข้อมูลลดน้อยลง

2. รักษาความถูกต้องของข้อมูล

เนื่องจากฐานข้อมูลมีเพียงฐานข้อมูลเดียว ใน กรณีที่มีข้อมูลชุดเดียวกันปรากฏอยู่หลายแห่งในฐานข้อมูล ข้อมูลเหล่านี้จะต้องตรงกัน ถ้ามีการแก้ไขข้อมูลนี้ทุก ๆ แห่งที่ข้อมูลปรากฏอยู่จะแก้ไขให้ถูกต้องตามกันหมดโดยอัตโนมัติด้วยระบบจัดการฐานข้อมูล

3. การป้องกันและรักษาความปลอดภัยให้กับข้อมูลทำได้อย่างสะดวก

การป้องกันและรักษาความปลอดภัยกับข้อมูลระบบฐานข้อมูลจะให้เฉพาะผู้ที่เกี่ยวข้องเท่านั้น ซึ่งก่อให้เกิดความปลอดภัย (security) ของข้อมูลด้วย

2.2 ด้านเทคโนโลยี

ในส่วนนี้จะอธิบายถึงเทคโนโลยีที่ใช้ในการพัฒนาเว็บแอปพลิเคชัน

2.2.1 HTML

HTML [4] ย่อมาจาก Hyper Text Markup Language คือภาษาคอมพิวเตอร์ที่ใช้ในการแสดงผลของเอกสารบน website หรือที่เราเรียกกันว่าเว็บเพจ ถูกพัฒนาและกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C) และจากการพัฒนาทางด้าน Software ของ Microsoft ทำให้ภาษา HTML เป็นอีกภาษาหนึ่งที่ใช้เขียนโปรแกรมได้ หรือที่เรียกว่า HTML Application HTML เป็นภาษาประเภท Markup สำหรับการการสร้างเว็บเพจ โดยใช้ภาษา HTML สามารถทำได้โดยใช้โปรแกรม Text Editor ต่างๆ เช่น VS Code, Vim หรือจะอาศัยโปรแกรมที่เป็นเครื่องมือช่วยสร้างเว็บเพจ เช่น Dream Weaver ซึ่งอำนวยความสะดวกในการสร้างหน้า HTML ส่วนการเรียกใช้งานหรือทดสอบการทำงานของเอกสาร HTML จะใช้โปรแกรม web browser เช่น Google Chrome, Microsoft Edge, Mozilla Firefox, Safari และ Opera เป็นต้น



รูปที่ 2.1: HTML

2.2.2 CSS

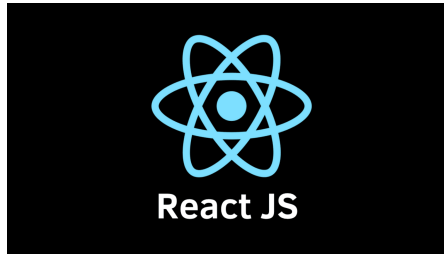
CSS [5] ย่อมาจาก Cascading Style Sheet มักเรียกโดยย่อว่า "สไตลชีต" คือภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS กำหนดกฎเกณฑ์ในการระบุรูปแบบ (หรือ "Style") ของเนื้อหาในเอกสาร อันได้แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งการกำหนดรูปแบบ หรือ Style นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสาร ไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลลัพท์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วกันทุกหน้าเอกสารภายในเว็บไซต์เดียวกัน โดยกฎเกณฑ์ในการกำหนดรูปแบบ (Style) เอกสาร HTML ถูกเพิ่มเข้ามาครั้งแรกใน HTML 4.0 เมื่อปีพ.ศ. 2539 ในรูปแบบของ CSS level 1 Recommendations ที่กำหนดโดย องค์กร World Wide Web Consortium หรือ W3C

2.2.3 TypeScript

Typescript [6] คือภาษา JavaScript ใน Version ที่ได้รับการ Upgrade สามารถทำงานบน Node.js Environment หรือ Web Browser ต่าง ๆ ที่มีการรองรับ ECMAScript 3 ขึ้นไป TypeScript เป็น Statically Compiled Language ที่ได้จัดเตรียมทั้ง Static Typing, Classes และ Interface ไว้ให้แล้ว ช่วยให้คุณสามารถเขียน Code ของ JavaScript ที่เรียบง่ายและ Clean ได้อย่างสะดวกขึ้น ดังนั้น การใช้ TypeScript จะช่วยให้คุณสร้าง Software ที่ปรับใช้งานได้ง่ายและมีประสิทธิภาพมากยิ่งขึ้น

2.2.4 React JS

React JS [7] เป็นไลบรารี JavaScript ที่ใช้สร้าง User Interface (UI) ในเว็บแอปพลิเคชันแบบ Single Page Application (SPA) และเว็บแอปพลิเคชันที่มีการอัปเดตสดๆ โดย React ช่วยให้ผู้พัฒนาสามารถสร้าง UI ที่มีประสิทธิภาพและเรียบง่ายได้ง่ายขึ้น ด้วยการใช้ Component-based architecture ที่ช่วยให้โค้ดสามารถรับมือกับขนาดของโปรเจกต์ได้ง่ายขึ้น และ การใช้ Virtual DOM ที่ช่วยเพิ่มประสิทธิภาพในการเปลี่ยนแปลง UI โดยไม่ต้องทำการอัปเดตทั้งหมดของ DOM



รูปที่ 2.2: React JS



รูปที่ 2.3: Mantine UI

ประโยชน์ของ React JS ได้แก่:

1. ประสิทธิภาพสูง: โดยใช้ Virtual DOM ทำให้การเปลี่ยนแปลง UI มีประสิทธิภาพและเร็วขึ้น
2. Component-based: ช่วยให้การจัดการ UI เป็นเรื่องง่ายและมีระเบียบมากขึ้น
3. Reusable Components: สามารถ reuse โค้ดของ Component ได้ซึ่งช่วยให้การพัฒนาเร็วขึ้น
4. รองรับการทำงานแบบ Server-side Rendering (SSR) และ Client-side Rendering (CSR): ทำให้เหมาะสำหรับการพัฒนาเว็บแอปพลิเคชันที่มี SEO ดีและประสิทธิภาพการโหลดที่ดี

2.2.5 Mantine UI

Mantine UI [8] เป็นไลบรารีของ React Component ที่ถูกออกแบบมาเพื่อช่วยในการพัฒนา UI ในโปรเจกต์ของเรา มันมีชุดของ Components ที่หลากหลายและนำใช้งานอย่างง่าย เหมาะสำหรับการสร้าง UI ที่สวยงามและมีประสิทธิภาพ การใช้ Mantine UI ช่วยให้เราสามารถประสานงานกับผู้ออกแบบและนักพัฒนาได้อย่างมีประสิทธิภาพเพื่อสร้างผลลัพธ์ที่มีคุณภาพและประทับใจได้ในโปรเจกต์ของเรา การนำ Mantine UI เข้ามาใช้ยังช่วยลดเวลาในการพัฒนาโดยทั่วไปด้วยความสามารถในการทดสอบและปรับแต่งที่มีอยู่อย่างสมบูรณ์ โดยทั้งหมดนี้ช่วยให้เราสามารถให้ผลลัพธ์สุดท้ายในโปรเจกต์ของเราได้อย่างมีประสิทธิภาพและมีคุณภาพสูงสุดในเวลาที่มีจำกัด

2.2.6 PostgreSQL

PostgreSQL [9] เป็นฐานข้อมูลเชิงสัมพันธ์แบบโอเพ่นซอร์สระดับธุรกิจที่ทรงพลัง อนุญาตให้ใช้ข้อมูลและแบบสอบถาม SQL เชิงสัมพันธ์และ JSON ที่ไม่ใช่เชิงสัมพันธ์ PostgreSQL มีชุมชนที่แข็งแกร่งอยู่เบื้องหลัง PostgreSQL เป็นระบบจัดการฐานข้อมูลที่น่าเชื่อถือมาก พร้อมการสนับสนุน ความปลอดภัย และ



รูปที่ 2.4: PostgreSQL

ความแม่นยำในระดับดีเยี่ยม โทรศัพท์มือถือและเว็บแอปพลิเคชันจำนวนมากใช้ PostgreSQL เป็นฐานข้อมูลเริ่มต้น โซลูชันเชิงพื้นที่และการวิเคราะห์จำนวนมากใช้ประโยชน์จาก PostgreSQL เวอร์ชันล่าสุดคือ PostgreSQL 15 PostgreSQL รองรับประเภทข้อมูลที่ซับซ้อน ในความเป็นจริง ฐานข้อมูลถูกสร้างขึ้นโดยคำนึงถึงประเภทข้อมูลจำนวนมาก ประสิทธิภาพของฐานข้อมูลนั้นใกล้เคียงกับของคู่แข่งเช่น Oracle และ SQL Server AWS ให้บริการฐานข้อมูลที่ได้รับการบำรุงรักษาอย่างสมบูรณ์สำหรับ PostgreSQL ด้วยบริการฐานข้อมูลเชิงสัมพันธ์ของ Amazon PostgreSQL ยังใช้ในการสร้าง Amazon Aurora อีกด้วย

คุณสมบัติที่สำคัญของ PostgreSQL

หนึ่งในเหตุผลที่ PostgreSQL ได้รับความนิยมมากเนื่องจากชุดคุณลักษณะ ฐานข้อมูลช่วยในการ พัฒนาแอปพลิเคชัน โดยการรักษาความสมบูรณ์ของข้อมูล ช่วยให้ผู้ใช้ดูแลระบบสามารถสร้างสภาพแวดล้อมที่ทนต่อความผิดพลาดได้ นอกจากนี้ยังสามารถใช้ข้ามแพลตฟอร์มที่หลากหลายและใช้ประโยชน์จากภาษาโปรแกรมทั่วไปได้ทั้งหมด เราจะเห็นรายชื่อที่แน่นอนในภายหลัง ฐานข้อมูลยังมีระบบล็อกขั้นสูงมาก นอกจากนี้ยังมีการควบคุมการทำงานพร้อมกันหลายเวอร์ชัน เซิร์ฟเวอร์ฐานข้อมูล PostgreSQL ยังมีฟังก์ชันการทำงานสำหรับการเขียนโปรแกรมฝั่งเซิร์ฟเวอร์สำหรับผู้ใหญ่อีกด้วย เป็นไปตามข้อกำหนด ANSI SQL และรองรับสถาปัตยกรรมเครือข่ายไคลเอ็นต์-เซิร์ฟเวอร์อย่างสมบูรณ์

PostgreSQL ยังมีความพร้อมใช้งานสูงและเซิร์ฟเวอร์สำรอง สอดคล้องกับ ANSI-SQL2008 และเชิงวัตถุ ความสามารถในการเชื่อมต่อกับคลังข้อมูลอื่นๆ เช่น NoSQL ซึ่งทำหน้าที่เป็นฮับแบบครบวงจรสำหรับระบบหลายภาษา สามารถทำได้ผ่านการสนับสนุน JSON ของฐานข้อมูล ข้อมูลของคลัสเตอร์ฐานข้อมูลเดียวจะได้รับการจัดการโดยอินสแตนซ์ PostgreSQL หนึ่งอินสแตนซ์เสมอ คลัสเตอร์ของฐานข้อมูลคือกลุ่มของเร็กคอร์ดที่เก็บไว้ในที่เดียวกันบนระบบไฟล์

2.2.7 Prisma

Prisma ORM [10] เป็นเครื่องมือช่วยในการจัดการฐานข้อมูลที่เป็น ORM (Object-Relational Mapping) ซึ่งถูกออกแบบมาเพื่อช่วยให้การเข้าถึงฐานข้อมูลเป็นไปอย่างมีประสิทธิภาพและง่ายตายตัวสำหรับผู้พัฒนาซอฟต์แวร์ โดย Prisma ORM ช่วยในการสร้าง query ที่ปลอดภัยและมีประสิทธิภาพอีกด้วย

ประโยชน์ของ Prisma ORM ได้แก่:

1. ความสะดวกและความยืดหยุ่นในการใช้งาน: Prisma ORM ช่วยลดความซับซ้อนในการจัดการฐานข้อมูลและทำให้การเขียนโค้ดเป็นเรื่องที่ง่ายขึ้น ด้วยรูปแบบที่เป็นตัวถามและเรียกใช้ method ของ Prisma เพื่อสร้างและจัดการข้อมูล



รูปที่ 2.5: Prisma ORM

2. การสร้าง query ที่ปลอดภัย: Prisma ORM ช่วยป้องกันการโจมตีด้านความปลอดภัยเช่น SQL Injection โดยมีการ validate และ escape ข้อมูลอัตโนมัติ
3. การสร้างฐานข้อมูลแบบพลวัต: Prisma ORM ช่วยให้สามารถสร้างและแก้ไขโครงสร้างของฐานข้อมูลได้อย่างง่ายดาย ผ่านการใช้งาน migration เพื่อเปลี่ยนแปลง schema ของฐานข้อมูล
4. ประสิทธิภาพและความเร็ว: Prisma ORM มีการจัดการข้อมูลแบบอัตโนมัติที่เป็นไปอย่างมีประสิทธิภาพ ทำให้การเข้าถึงข้อมูลเป็นไปอย่างรวดเร็ว

Prisma ORM ได้ถูกพัฒนาโดย Prisma Labs และเป็นโปรเจกต์โอเพนซอร์ส ที่สามารถใช้งานได้ฟรีและเป็นส่วนหนึ่งของชุมชนนักพัฒนาซอฟต์แวร์ สามารถดาวน์โหลดและใช้งาน Prisma ORM ได้จากเว็บไซต์ของเครื่องมือนี้แบบฟรี นอกจากนี้ยังมีเอกสารและคู่มือการใช้งานที่มีอยู่ในชุมชนสำหรับการศึกษาและการใช้งานอื่น ๆ อีกด้วย

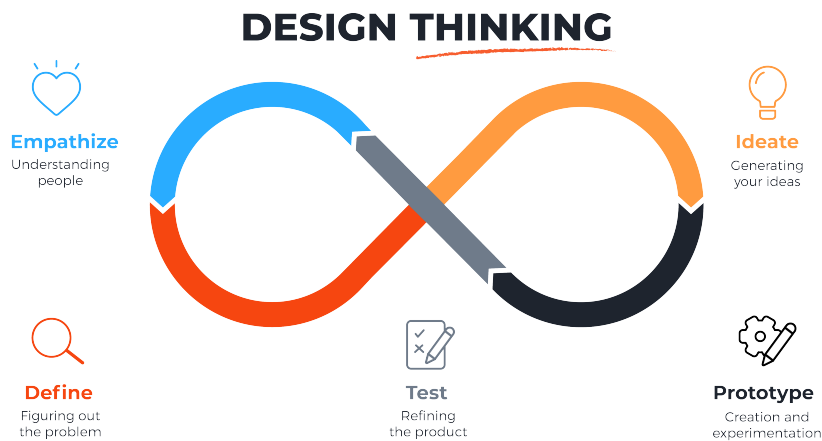
2.3 ด้าน User Interface

ในส่วนนี้จะอธิบายถึงการออกแบบ User Interface ของเว็บแอปพลิเคชัน

2.3.1 Design Thinking

กระบวนการออกแบบ design thinking นั้นมีหลากหลายรูปแบบ ทั้งรูปแบบ 3 ชั้น ไปจนถึง 7 ชั้น ทุกรูปแบบมีความคล้ายคลึงมากที่สุด และใช้หลักการเดียวกันที่อ้างอิงจาก Herbert Simon ผู้ชนะรางวัลโนเบลในสาขา The Sciences of the Artificial ในปี 1969 โดยรูปแบบที่นิยมใช้กันมากที่สุด คือ รูปแบบของ Hasso-Plattner Institute of Design at Stanford มีทั้งหมด 5 กระบวนการด้วยกัน ดังนี้

1. Empathise หรือ การเข้าใจปัญหา คือ การทำความเข้าใจกับปัญหาก่อน ตั้งแต่การเข้าใจผู้ใช้ กลุ่มเป้าหมาย หรือเข้าใจสิ่งที่ต้องการแก้ไขเพื่อหาหนทางที่เหมาะสม และดีที่สุดให้ได้ โดยเริ่มต้นจากการเข้าใจคำถาม สร้างสมมติฐาน กระตุ้นให้เกิดการใช้ความคิดที่นำไปสู่ความคิด สร้างสรรค์ และวิเคราะห์ปัญหาให้ถี่ถ้วน เพื่อหาแนวทางที่ชัดเจน นำไปสู่การแก้ไขปัญหาที่ตรงประเด็น และสร้างผลลัพธ์ที่ดีที่สุด



รูปที่ 2.6: กระบวนการออกแบบ Design Thinking

2. **Define** หรือ กำหนดปัญหาให้ชัดเจน คือ การเข้าใจความต้องการ ปัญหา และวิเคราะห์ข้อมูลเชิงลึก เพื่อคัดกรองหาปัญหาที่แท้จริง กำหนดหรือบ่งชี้ปัญหาอย่างชัดเจน เพื่อที่จะเป็นแนวทางในการปฏิบัติ และมีทิศทางในการแก้ไขปัญหาอย่างชัดเจน
3. **Ideate** หรือ ระดมความคิด คือ การนำเสนอแนวคิดต่างๆร่วมกัน ถึงวิธีการแก้ไขปัญหา อย่างไม่มีกรอบจำกัด การระดมความคิดควรมีมุมมองหลากหลาย และมีหลากหลายแนวทางให้ได้มากที่สุด เพื่อให้มีฐานข้อมูลในการนำไปวิเคราะห์และสรุปผล เพื่อนำไปแก้ไขปัญหา โดยไม่จำเป็นต้องเป็นแนวทางใดแนวทางหนึ่ง และการระดมความคิดยังช่วยมองเห็นปัญหาที่หลากหลายได้มากขึ้น
4. **Prototype** หรือ สร้างต้นแบบที่เลือก คือ การออกแบบผลิตภัณฑ์หรือนวัตกรรม เพื่อสร้างต้นแบบสำหรับการทดสอบ และนำไปใช้จริง ซึ่งคือ การลงมือปฏิบัติหรือทดลองตามแนวทางการแก้ไขปัญหาที่ได้กำหนดไว้
5. **Test** หรือ ทดสอบการแก้ไขปัญหา นำ **Prototype** ที่เราทำการทำขึ้นมาไปทดสอบกับผู้ใช้ว่าสามารถแก้ไขปัญหาของ ผู้ใช้ได้หรือไม่ และหลังจากนั้นถ้าหากการแก้ปัญหายังไม่สามารถช่วยแก้ไขได้ หรือแก้ไขได้ยังไม่ดีพอ ผู้จัดทำจะต้องกลับไปทำตั้งแต่ขั้นตอนแรกอีกครั้งจนกว่าจะสามารถออกแบบโปรแกรมที่แก้ไขปัญหา ของผู้ใช้ได้

2.4 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ

- 261207 Basic CPE Lab นำความรู้ทางด้านการพัฒนาเว็บแอปพลิเคชัน เช่น HTML, CSS, Tailwind CSS, JavaScript, TypeScript, Next.js และ Node.js มาประยุกต์ใช้ในการพัฒนาเว็บแอปพลิเคชัน ทั้งด้านของ front-end ซึ่งจะแสดงผลของเว็บไซต์ และ back-end ที่จะจัดการการทำงานต่าง ๆ รวมถึงการเชื่อมต่อกับฐานข้อมูล
- 261361 Software Engineering การใช้กระบวนการทางวิศวกรรมในการดูแลการผลิต ตั้งแต่การเริ่มเก็บความต้องการ การตั้งเป้าหมายของระบบ การออกแบบ กระบวนการพัฒนา การตรวจสอบ การประเมินผลและทดสอบระบบ

- 261346 Database Systems การใช้งานฐานข้อมูล โดยใช้ PostgreSQL ในการจัดการฐานข้อมูล รวมถึงการเชื่อมต่อกับฐานข้อมูล และการเขียนคำสั่ง SQL ในการดึงข้อมูลจากฐานข้อมูลและการจัดการข้อมูลในฐานข้อมูล

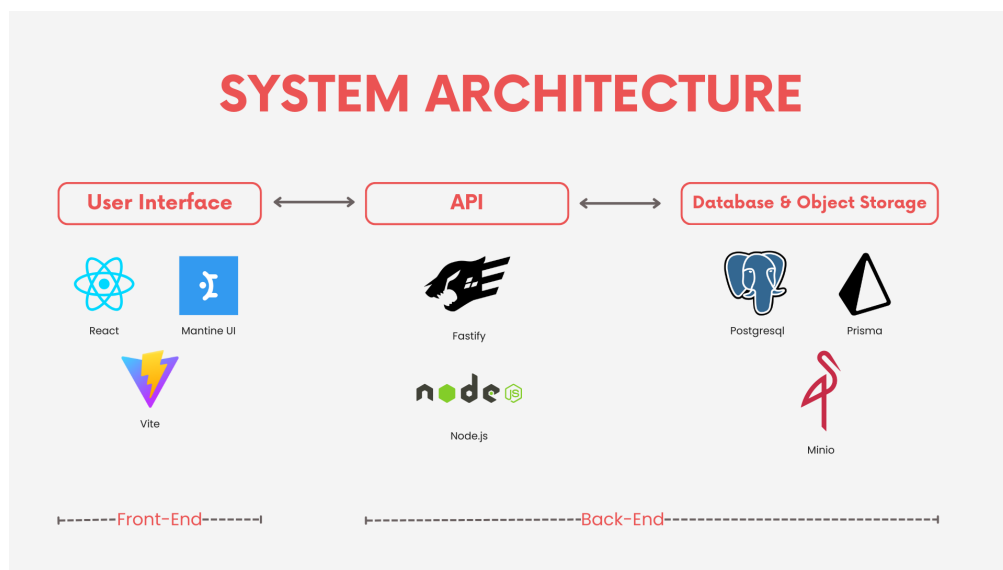
บทที่ 3

โครงสร้างและขั้นตอนการทำงาน

ในบทนี้จะกล่าวถึงหลักการ และการออกแบบระบบ

3.1 หลักการทำงานของระบบ

3.1.1 การทำงานของระบบ (System Architecture)



รูปที่ 3.1: System Architecture

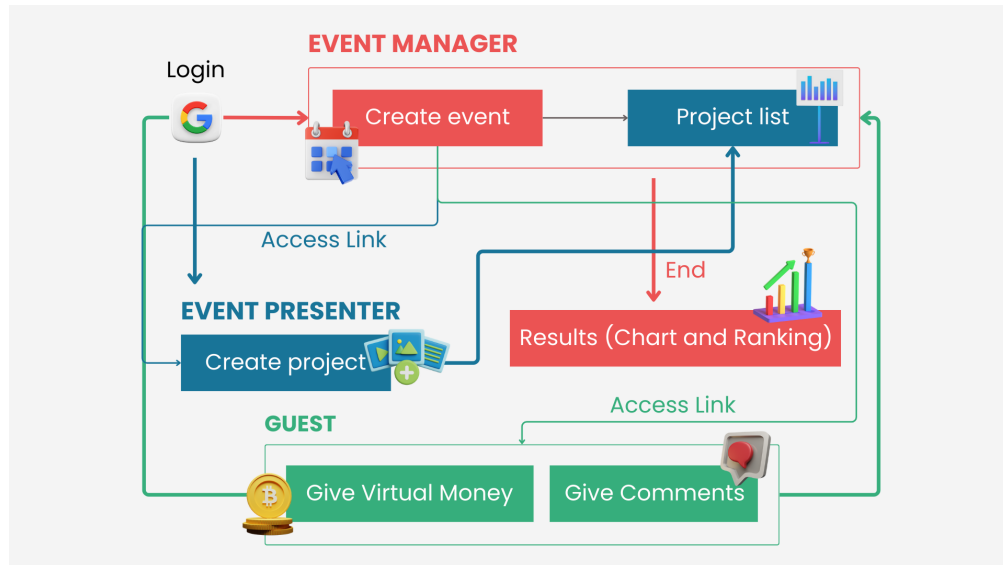
จากรูปที่ 3.1 จะเป็นภาพรวมของระบบที่เราได้ทำการออกแบบขึ้น โดยมีรายละเอียดดังนี้

- Frontend

ส่วนหน้าบ้าน (Frontend) เป็นส่วนการพัฒนาเพื่อแสดง User Interface (UI) โดยโครงการนี้ได้ใช้เทคโนโลยี React ร่วมกับ Mantine UI ในการออกแบบและสร้างส่วนติดต่อผู้ใช้ (UI) โดยมีหน้าที่แสดงผลต่อผู้ใช้ในรูปแบบที่เข้าใจง่าย รับข้อมูลป้อนจากผู้ใช้ผ่านอินเทอร์เฟซต่าง ๆ เช่น ปุ่ม ฟิลด์ข้อความ ฯลฯ และสื่อสารกับ API เพื่อส่งคำร้องขอและรับผลลัพธ์

- Backend

ส่วนหลังบ้าน (Backend) เป็นส่วนที่ทำหน้าที่รับข้อมูลจากผู้ใช้ จากนั้นทำการประมวลผลข้อมูล และส่งข้อมูลกลับไปยังผู้ใช้ โดยโครงการนี้ได้ใช้เทคโนโลยี Fastify เว็บเฟรมเวิร์ค Node.js ร่วมกับ Prisma ORM ในการจัดการข้อมูลในฐานข้อมูล รวมถึงทำการสื่อสารกับฐานข้อมูล PostgreSQL และ Minio (Object Storage) ในการจัดการข้อมูลที่เป็นไฟล์



รูปที่ 3.2: User Flow

3.1.2 เส้นทางของผู้ใช้ (User Flow)

จากรูปที่ 3.2 จะเป็นเส้นทางของผู้ใช้งานที่เข้าใช้งานระบบ โดยมีรายละเอียดดังนี้

ผู้สร้างงานกิจกรรม (Event Manager)

- ผู้ใช้เข้าสู่ระบบโดยใช้ Google Account
- ผู้ใช้สร้าง Event ใหม่ โดยป้อนข้อมูลต่างๆ เช่น ชื่องานกิจกรรม รายละเอียด วันเวลา สถานที่ เป็นต้น
- ผู้ใช้สร้าง Event เสร็จสิ้น จะมี QR Code หรือ Access Link สำหรับนำไปให้ผู้นำเสนอโครงการ (Presenter) สำหรับเพิ่มโครงการของตนและผู้เข้าร่วมกิจกรรม (Guest) สำหรับเข้าร่วมกิจกรรม
- ผู้ใช้สามารถดูผลลัพธ์ (Chart และ Ranking) ของโครงการต่าง ๆ ที่เข้าร่วมกิจกรรมในระหว่างจัดงานกิจกรรม

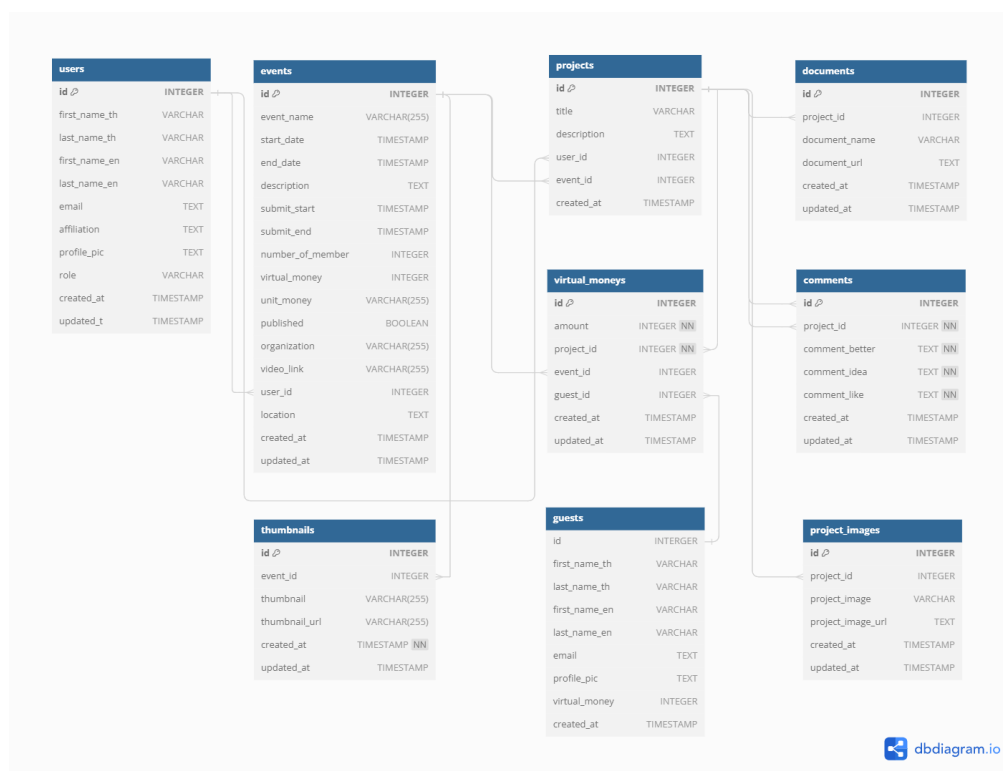
ผู้นำเสนอโครงการ (Presenter)

- ผู้ใช้เข้าสู่ระบบโดยใช้ Google Account
- ผู้ใช้สร้าง Project ใหม่ โดยป้อนข้อมูลต่าง ๆ เช่น ชื่อโครงการ รายละเอียด รูปภาพ ลิงก์ และไฟล์อื่น ๆ ที่เกี่ยวข้อง
- ผู้ใช้ดูผลลัพธ์ว่าโครงการของตนได้รับ Virtual Money จากผู้เข้าร่วมกิจกรรม (Guest) และความคิดเห็นเกี่ยวกับโครงการของตนหลังจากเสร็จสิ้นงานกิจกรรม

ผู้เข้าร่วมกิจกรรม (Guest)

- ผู้ใช้เข้าร่วม Event โดยใช้ QR CODE หรือ Access Link ที่ได้รับจาก Event Manager
- ผู้ใช้เข้าสู่ระบบโดยใช้ Google Account
- ผู้ใช้ดูรายละเอียดของงานกิจกรรมที่จัดขึ้น รวมถึงโครงการที่เข้าร่วมกิจกรรม
- ผู้ใช้สามารถให้ Virtual Money และแสดงความคิดเห็นเกี่ยวกับ Project ต่าง ๆ ที่เข้าร่วมกิจกรรมได้

3.1.3 โครงสร้างฐานข้อมูล (Database Schema)



รูปที่ 3.3: Database Schema

จากรูปที่ 3.3 จะเป็นโครงสร้างของฐานข้อมูลที่ใช้ในระบบ โดยมีรายละเอียดดังนี้

3.2 ส่วนเชื่อมต่อระหว่างผู้ใช้งานกับระบบ (User Interface)

3.2.1 ผู้ใช้งาน (User)

หน้าเข้าสู่ระบบ (Login Page)

Artificial Intelligence Detecting Oral Cancer

ระบบการวิเคราะห์โรคด้วย AI

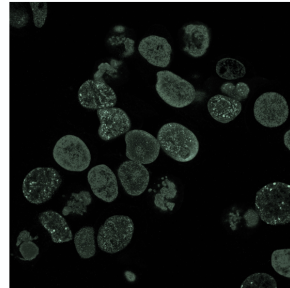
ชื่อผู้ใช้งาน

รหัสผ่าน

☐ ลืมรหัสผ่านของคุณ

[ลืมรหัสผ่าน ?](#)

เข้าสู่ระบบ



AI V.1.10

AI ในการประมวลผลรูปภาพ

WEB V.1.10

Website ในการรองรับข้อมูลรูปภาพ

200+

จำนวนโรงพยาบาลที่ใช้งาน

100+

จำนวนผู้ใช้งานทั่วไป

คำชี้แจงและเงื่อนไขการใช้งาน

Lorem ipsum dolor sit amet consectetur. Venenatis odio lectus eget senectus felis sed ornare elit proin. Nunc venenatis faucibus enim adipiscing. Diam maecenas dapibus purus praesent quam quis mattis venenatis.

#1

Lorem ipsum dolor sit amet consectetur. In nisl feugiat at tempus sodales amet vel netus at. Enim et sodales cursus metus mauris ullamcorper. Ipsum id ultricies iaculis vel sed. Eu non dolor accumsan quisque faucibus blandit morbi praesent.

#2

Lorem ipsum dolor sit amet consectetur. In nisl feugiat at tempus sodales amet vel netus at. Enim et sodales cursus metus mauris ullamcorper. Ipsum id ultricies iaculis vel sed. Eu non dolor accumsan quisque faucibus blandit morbi praesent.



ชื่อผู้

Lorem ipsum dolor sit amet consectetur adipiscing elit.

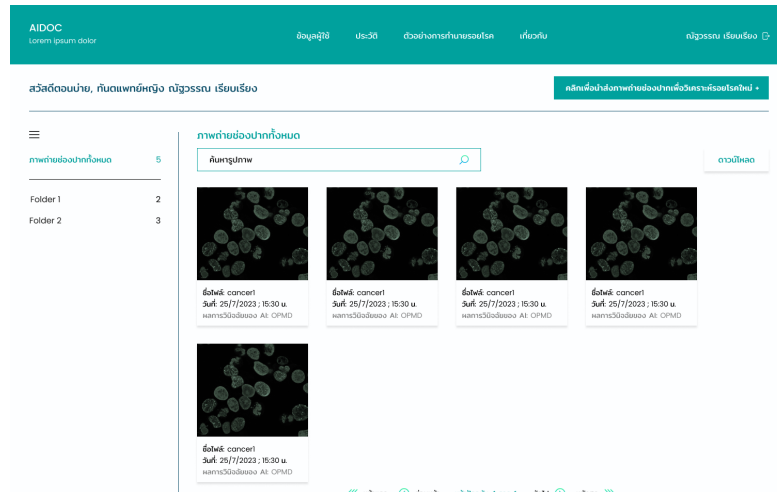
ติดต่อ

Lorem ipsum dolor sit amet consectetur adipiscing elit.

รูปที่ 3.4: Login Page

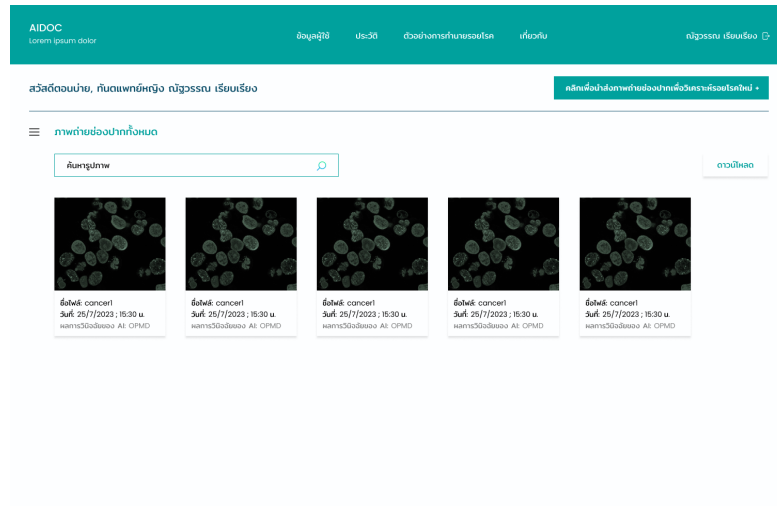
หน้าลงทะเบียน (Register Page)

หน้าประวัติการอัปโหลดรูปภาพ (History Page)



รูปที่ 3.6: History Page

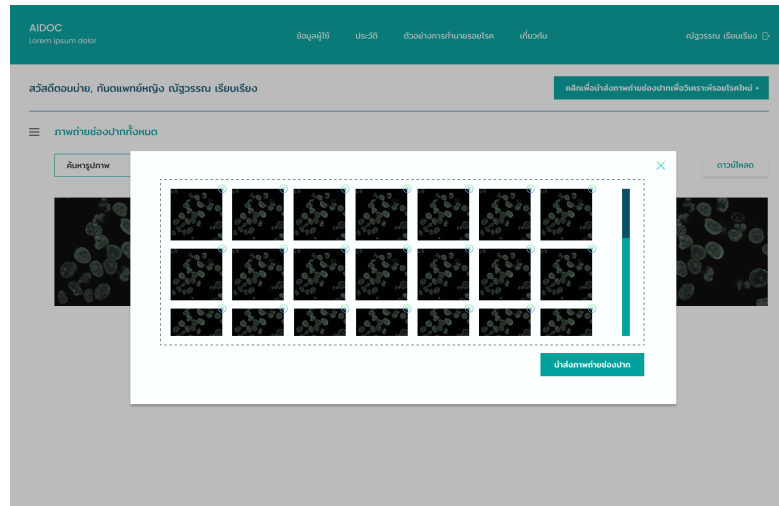
หน้าแสดงผลการทำนายรอยโรคในช่องปาก (Result Page)



รูปที่ 3.7: History Page

หน้าสำหรับตรวจรอยโรคในช่องปาก (Annotation Page)

หน้าแก้ไขข้อมูลส่วนตัว (Edit Profile Page)

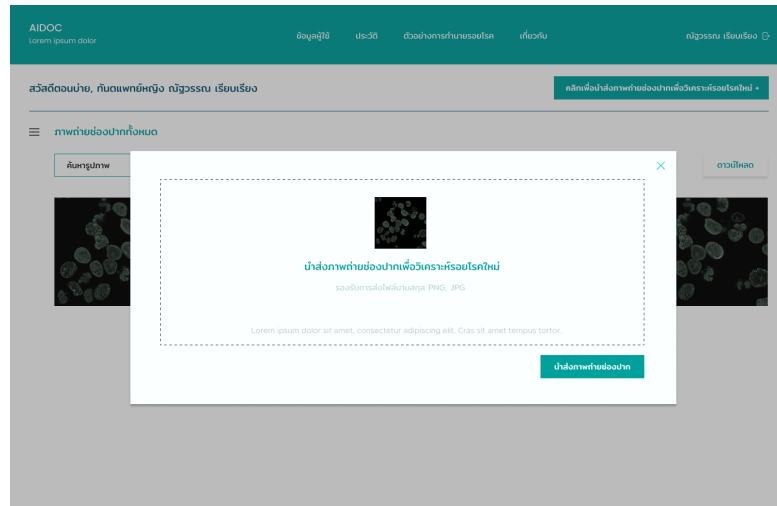


รูปที่ 3.8: Upload Page

3.2.2 ผู้ดูแลระบบ (Admin)

หน้าอัปโหลดรูปภาพทั้งหมด (Upload Page)

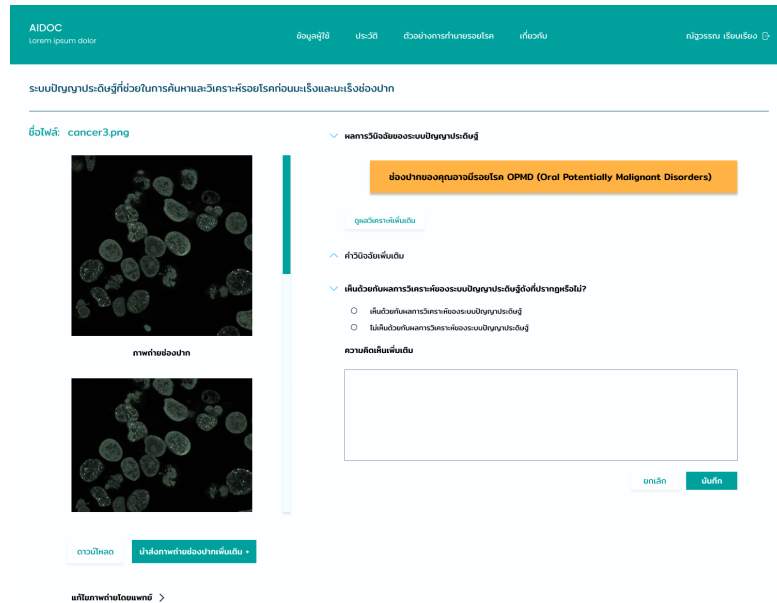
หน้าจัดการผู้ใช้งาน (User Management Page)



รูปที่ 3.9: Upload Page

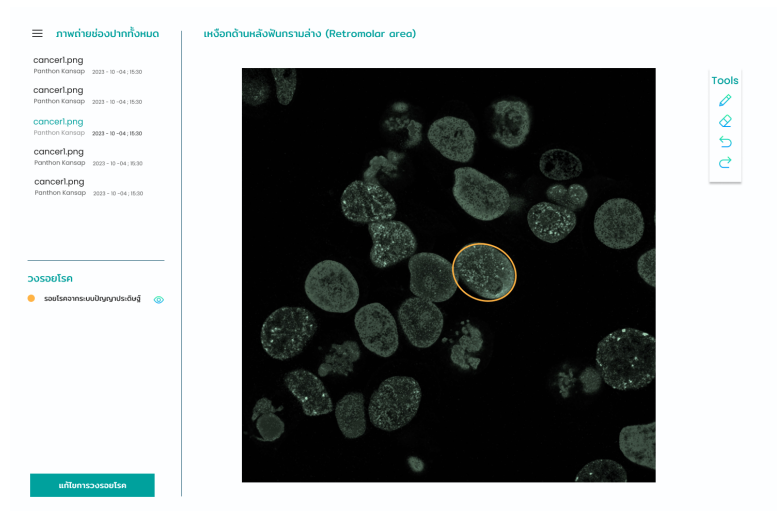
หน้าจัดการข้อมูล (Data Management Page)

หน้าจัดการข้อมูล (Data Management Page) โดยแยกตามผู้ใช้งาน

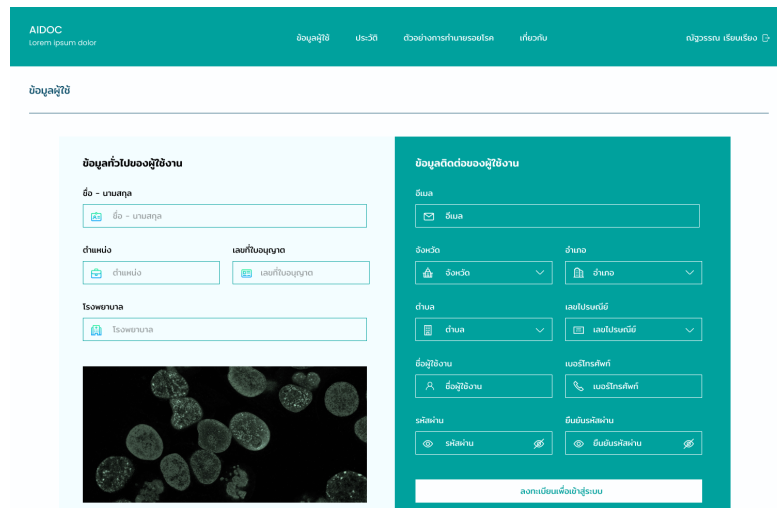


รูปที่ 3.10: Result Page

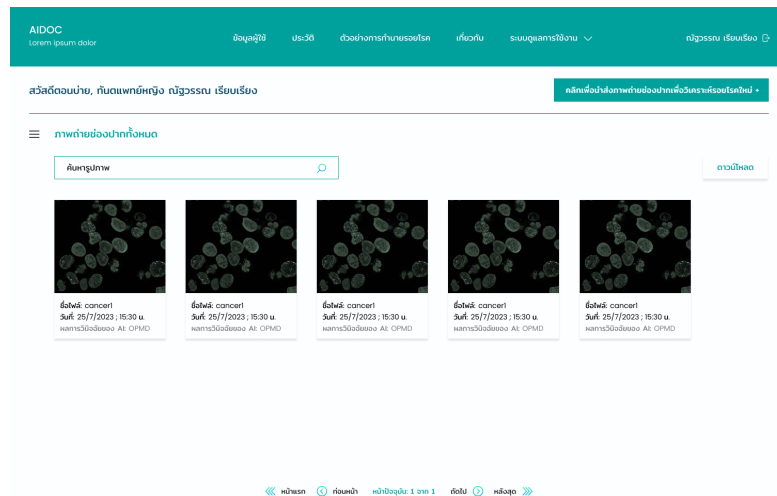
หน้าจัดการผู้ใช้งาน (User Management Page) โดยแยกตามผู้ใช้งาน



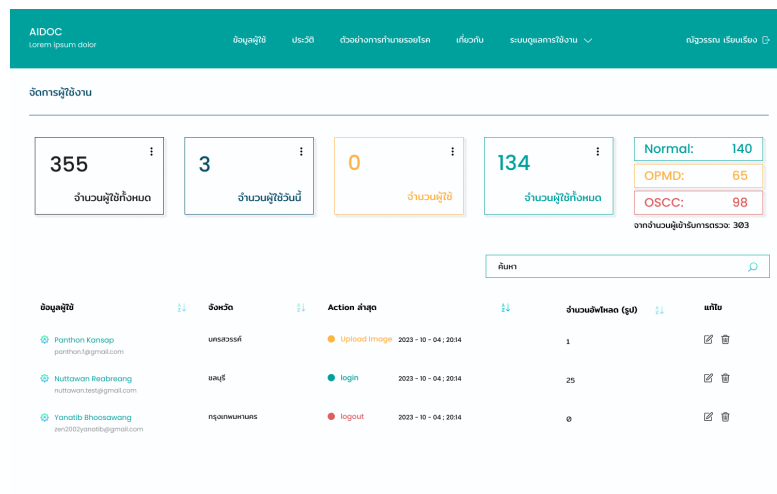
รูปที่ 3.11: Annotation Page



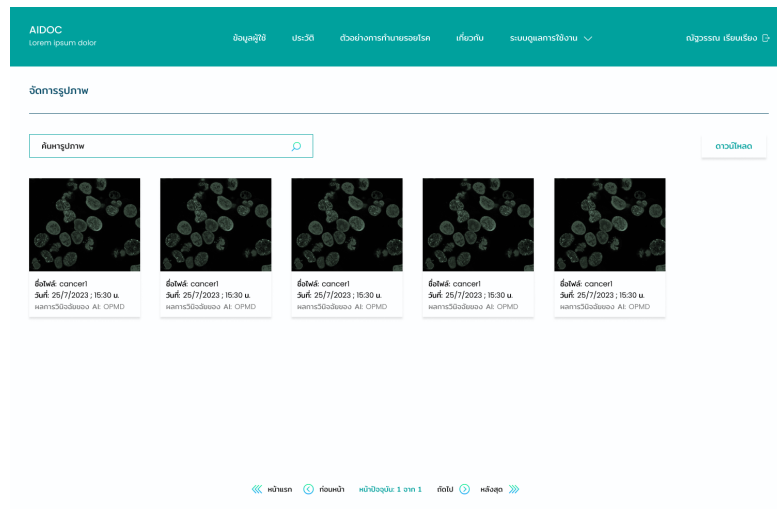
รูปที่ 3.12: Edit Profile Page



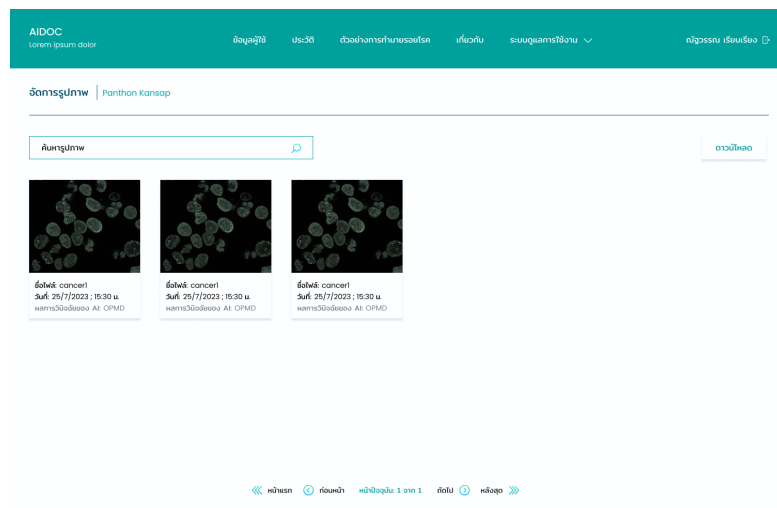
รูปที่ 3.13: Upload Page



รูปที่ 3.14: User Management Page



รูปที่ 3.15: Data Management Page



รูปที่ 3.16: Data Management Page

ข้อมูลทั่วไปของผู้ใช้งาน

ชื่อ - นามสกุล

ชื่อ - นามสกุล

ตำแหน่ง

ตำแหน่ง

สถานที่ในหน่วยงาน

สถานที่ในหน่วยงาน

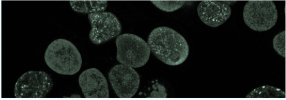
โทรศัพท์มือถือ

โทรศัพท์มือถือ

นามสกุล

User

Admin



ข้อมูลติดต่อของผู้ใช้งาน

อีเมล

อีเมล

จังหวัด

จังหวัด

อำเภอ

อำเภอ

ตำบล

ตำบล

เลขที่โทรศัพท์

เลขที่โทรศัพท์

ชื่อผู้ใช้งาน

ชื่อผู้ใช้งาน

รหัสผ่าน

รหัสผ่าน

ยืนยันรหัสผ่าน

ยืนยันรหัสผ่าน

บันทึกการเปลี่ยนแปลง

รูปที่ 3.17: Data Management Page

บทที่ 4

การทดลองและผลลัพธ์

ในบทนี้จะทดสอบเกี่ยวกับการทำงานในฟังก์ชันหลัก ๆ

4.1 การทดลองเกี่ยวกับการทำงานของระบบ

การประเมินระบบจะประเมินโดยทดสอบกับกลุ่มผู้ใช้งานทั้ง 4 กลุ่ม ได้แก่ ผู้ใช้ทั่วไป, ทันตแพทย์, ทันตบุคลากร และอาสาสมัครสาธารณสุขประจำหมู่บ้าน (อสม.) โดยในการทดสอบระบบจะมีการประเมินผลการทดลองโดยใช้เกณฑ์ต่าง ๆ ดังนี้

4.1.1 ผู้ใช้ทั่วไป

ผู้ใช้ทั่วไปก็มีความต้องการใช้งานระบบที่เรียบง่าย ใช้งานง่าย ไม่ซับซ้อน ดังนั้น ในการทดสอบกับผู้ใช้ทั่วไป ควรเน้นการประเมินปัจจัยต่างๆ เช่น

- ความน่าใช้งาน: Ease of use
- ความพึงพอใจของผู้ใช้งาน: User satisfaction
- ประโยชน์: Benefits

ตัวอย่างวิธีการทดสอบกับผู้ใช้ทั่วไป ได้แก่

- ให้ผู้ใช้งานทดสอบระบบและรวบรวมข้อมูลเกี่ยวกับประสบการณ์การใช้งาน เช่น ระยะเวลาในการดำเนินการแต่ละขั้นตอน ความสะดวกในการใช้งาน เป็นต้น
- ให้ผู้ใช้งานตอบแบบสอบถามเกี่ยวกับความพึงพอใจต่อระบบ เช่น ความง่ายในการใช้งาน ความน่าสนใจของเนื้อหา เป็นต้น
- ให้ผู้ใช้งานประเมินประโยชน์ที่ได้รับจากระบบ เช่น ช่วยให้ประหยัดเวลา ช่วยให้เข้าใจข้อมูลต่างๆ ได้ง่ายขึ้น เป็นต้น

4.1.2 ทันตแพทย์

ทันตแพทย์มีความต้องการใช้งานระบบที่มีประสิทธิภาพ ถูกต้องแม่นยำและสามารถช่วยในตรวจคัดกรองมะเร็งช่องปากได้อย่างมีประสิทธิภาพ ดังนั้น ในการทดสอบกับทันตแพทย์ ควรเน้นการประเมินปัจจัยต่างๆ เช่น

- ความน่าใช้งาน: Ease of use
- ความพึงพอใจของผู้ใช้งาน: User satisfaction
- ประโยชน์: Benefits
- การช่วยในการตรวจคัดกรองมะเร็งช่องปาก: Screening

ตัวอย่างวิธีการทดสอบกับทันตแพทย์ ได้แก่

- ให้ทันตแพทย์ทดสอบระบบภายใต้สถานการณ์จริง เช่น ถ่ายภาพช่องปากของผู้ป่วย และให้ระบบตรวจคัดกรอง และให้ทันตแพทย์ประเมินความถูกต้องแม่นยำของระบบ เป็นต้น
- ให้ทันตแพทย์ประเมินประโยชน์ที่ได้รับจากระบบ เช่น ช่วยให้ตรวจคัดกรองมะเร็งช่องปากได้อย่างมีประสิทธิภาพหรือไม่ เป็นต้น

4.1.3 ทันทบุคลการ

ทันทบุคลการมีความต้องการใช้งานระบบที่อำนวยความสะดวกในการทำงาน เช่น การดูประวัติการตรวจคัดกรองมะเร็งช่องปาก การบันทึกข้อมูล การสรุปผลการตรวจคัดกรองมะเร็งช่องปาก ดังนั้น ในการทดสอบกับทันทบุคลการ ควรเน้นการประเมินปัจจัยต่างๆ เช่น

- ความสะดวกในการใช้งาน: Ease of use
 - ประโยชน์: Benefits
- ตัวอย่างวิธีการทดสอบกับทันทบุคลการ ได้แก่
- ให้ทันทบุคลการทดสอบระบบและรวบรวมข้อมูลเกี่ยวกับประสบการณ์การใช้งาน เช่น ระยะเวลาในการดำเนินการแต่ละขั้นตอน ความสะดวกในการใช้งาน เป็นต้น
 - ให้ทันทบุคลการประเมินประโยชน์ที่ได้รับจากระบบ เช่น ระบบช่วยให้ทำงานได้อย่างมีประสิทธิภาพหรือไม่ เป็นต้น

4.1.4 อาสาสมัครสาธารณสุขประจำหมู่บ้าน (อสม.)

อสม. มีความต้องการใช้งานระบบที่เข้าใจง่าย ใช้งานสะดวก และสามารถช่วยให้ให้บริการประชาชนได้อย่างมีประสิทธิภาพ ดังนั้น ในการทดสอบกับอสม. ควรเน้นการประเมินปัจจัยต่าง ๆ เช่น

- ความน่าใช้งาน: Ease of use
 - ความพึงพอใจของผู้ใช้งาน: User satisfaction
 - ประโยชน์: Benefits
- ตัวอย่างวิธีการทดสอบกับอสม. ได้แก่
- ให้อสม.ทดสอบระบบและรวบรวมข้อมูลเกี่ยวกับประสบการณ์การใช้งาน เช่น ระยะเวลาในการดำเนินการแต่ละขั้นตอน ความสะดวกในการใช้งาน เป็นต้น
 - ให้อสม.ตอบแบบสอบถามเกี่ยวกับความพึงพอใจต่อระบบ เช่น ความง่ายในการใช้งาน ความน่าสนใจของเนื้อหา เป็นต้น
 - ให้อสม.ประเมินประโยชน์ที่ได้รับจากระบบ เช่น ระบบช่วยให้ให้บริการประชาชนได้อย่างมีประสิทธิภาพหรือไม่ เป็นต้น

ทั้งนี้ ในการทดสอบระบบกับผู้ทั่วไป, ทันตแพทย์, ทันทบุคลการและอสม. จะพิจารณาจากปัจจัยต่าง ๆ เช่น วัตถุประสงค์ของการประเมิน ขอบเขตของการประเมิน ความพร้อมของระบบ เป็นต้น เพื่อให้ได้ผลการประเมินที่มีประสิทธิภาพ

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผล

นศ. ควรสรุปถึงข้อจำกัดของระบบในด้านต่างๆ ที่ระบบมีในเนื้อหาส่วนนี้ด้วย

5.2 ปัญหาที่พบและแนวทางการแก้ไข

ในการทำโครงงานนี้ พบว่าเกิดปัญหาหลักๆ ดังนี้

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

ข้อเสนอแนะเพื่อพัฒนาโครงงานนี้ต่อไป มีดังนี้

บรรณานุกรม

- [1] CodeBee. (2016) Mvc คืออะไร ทำความเข้าใจรูปแบบการเขียนโปรแกรม. เข้าชมล่าสุดเมื่อ: 2 ตุลาคม 2566. [Online]. Available: <https://www.codebee.co.th/labs/mvc-คืออะไร-ทำความเข้าใจ/>
- [2] A. W. S. (AWS). (2023) Restful api คืออะไร. เข้าชมล่าสุดเมื่อ: 2 ตุลาคม 2566. [Online]. Available: <https://aws.amazon.com/th/what-is/restful-api/>
- [3] mindphp.com. (2022) Database คืออะไร ระบบฐานข้อมูล คือ การเก็บรวบรวมข้อมูลไว้ด้วยกัน อย่างมีระบบ. เข้าชมล่าสุดเมื่อ: 2 ตุลาคม 2566. [Online]. Available: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2055-database-คืออะไร23.html>
- [4] ———. (2022) Html คืออะไร เอชทีเอ็มแอล ภาษาคอมพิวเตอร์ที่ใช้ในการสร้างเว็บเพจ ใช้เขียนโปรแกรม ย่อมาจากอะไร. เข้าชมล่าสุดเมื่อ: 2 ตุลาคม 2566. [Online]. Available: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2026-html-คืออะไร.html>
- [5] wynnsoft solution. (2016) Css คืออะไร ? มีประโยชน์อย่างไรบ้าง. เข้าชมล่าสุดเมื่อ: 3 ตุลาคม 2566. [Online]. Available: <https://www.wynnsoft-solution.net/th/article/view/80/>
- [6] S. Damyos. (2022) Typescript คืออะไร. เข้าชมล่าสุดเมื่อ: 3 ตุลาคม 2566. [Online]. Available: <https://il.mahidol.ac.th/th/i-learning-clinic/computer-articles/ทำไมถึงต้องใช้-typescript-ทั้งที่/>
- [7] R. Documentation. (2023) React - a javascript library for building user interfaces. เข้าชมล่าสุดเมื่อ: 3 ตุลาคม 2566. [Online]. Available: <https://reactjs.org/>
- [8] V. Rtishchev. (2023) Mantine ui. เข้าชมล่าสุดเมื่อ: 3 กรกฎาคม 2566. [Online]. Available: <https://mantine.dev/>
- [9] <https://appmaster.io/th/blog/postgresql-khuue-aair>. (2022) Postgresql คืออะไร. เข้าชมล่าสุดเมื่อ: 3 ตุลาคม 2566. [Online]. Available: <https://appmaster.io/th/blog/postgresql-khuue-aair>
- [10] Prisma. (2023) Prisma | next-generation orm for node.js and typescript. เข้าชมล่าสุดเมื่อ: 3 ตุลาคม 2566. [Online]. Available: <https://www.prisma.io/>

ภาคผนวก

ภาคผนวก ก

The first appendix

Text for the first appendix goes here.

ก.1 Appendix section

Text for a section in the first appendix goes here.

test ทดสอบฟอนต์ serif ภาษาไทย

test ทดสอบฟอนต์ sans serif ภาษาไทย

test ทดสอบฟอนต์ teletype ภาษาไทย

test ทดสอบฟอนต์ teletype ภาษาไทย

ตัวหนา serif ภาษาไทย **sans serif ภาษาไทย teletype ภาษาไทย**

ตัวเอียง *serif ภาษาไทย sans serif ภาษาไทย teletype ภาษาไทย*

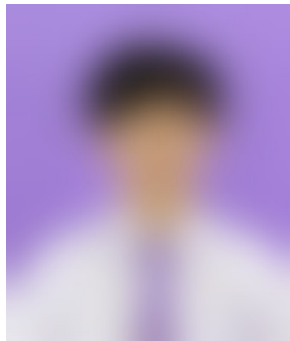
ตัวหนาเอียง ***serif ภาษาไทย sans serif ภาษาไทย teletype ภาษาไทย***

https://www.example.com/test_ทดสอบ_url

ภาคผนวก ข
คู่มือการใช้งานระบบ

Manual goes here.

ประวัติผู้เขียน



Your biosketch goes here. Make sure it sits inside the biosketch environment.