

โครงการเลขที่ วศ.คพ. S809-1/2566

เรื่อง

แพลตฟอร์มดิจิทัลสำหรับการตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและ
มะเร็งช่องปาก

โดย

นายญาณธิป ภู่อว่าง รหัส 630612097

นายปิ่นธีร กัณฑ์พย์ รหัส 630612105

รายงานนี้เป็นส่วนหนึ่งของวิชาสำรวจเพื่อโครงการ
ตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2566

PROJECT No. CPE S809-1/2566

**Digital Platform for Detecting and Analyzing Oral Potentially Malignant
Disorders and Oral Cancer**

Yanatip Bhoosawang 630612097

Panthon Kansap 630612105

**A Report Submitted in Partial Fulfillment of Project Survey Course
as Required by the Degree of Bachelor of Engineering**

Department of Computer Engineering

Faculty of Engineering

Chiang Mai University

2023

หัวข้อโครงการ : แพลตฟอร์มดิจิทัลสำหรับการตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็ง
และมะเร็งช่องปาก
: Digital Platform for Detecting and Analyzing Oral Potentially Malignant
Disorders and Oral Cancer
โดย : นายญาณธิป ภู่ว่าง รหัส 630612097
นายปิ่นภัทร กันทรพย์ รหัส 630612105
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : ผศ.ดร. ปฏิวธ วุฒิสารวัฒนา
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2566

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(ศ.ดร. สันติ พิทักษ์กัญญกุล)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ
(ผศ.ดร. ปฏิวธ วุฒิสารวัฒนา)

..... กรรมการ
(รศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล)

..... กรรมการ
(ผศ. โดม โพธิ์กานนท์)

สารบัญ

สารบัญ	ข
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	2
1.7 บทบาทและความรับผิดชอบ	2
2 ทฤษฎีที่เกี่ยวข้อง	3
2.1 ด้านโครงสร้างเว็บแอปพลิเคชัน	3
2.1.1 MVC Architecture	3
2.1.2 RESTful API	3
2.1.3 ระบบฐานข้อมูล (Database System)	4
2.2 ด้านเทคโนโลยี	5
2.2.1 HTML	5
2.2.2 CSS	5
2.2.3 TypeScript	6
2.2.4 Tailwind CSS	6
2.2.5 Next.js	7
2.2.6 MySQL	7
2.2.7 JSON	8
2.3 ด้าน User Interface	8
2.3.1 Design Thinking	8
2.4 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ	9
2.5 ความรู้นอกหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ	10
3 โครงสร้างของโครงการ	11
3.1 หลักการทำงานของระบบ	11
3.1.1 ภาพรวมของระบบ (System Overview)	11
3.1.2 โครงสร้างฐานข้อมูล (Database Schema)	11
3.2 ส่วนเชื่อมต่อระหว่างผู้ใช้งานกับระบบ (User Interface)	13
3.2.1 ผู้ใช้งาน (User)	13
3.2.2 ผู้ดูแลระบบ (Admin)	18
4 การประเมินระบบ	21
4.1 การทดลองเกี่ยวกับการทำงานของระบบ	21
4.1.1 ผู้ใช้ทั่วไป	21
4.1.2 ท้นดแพทย์	21
4.1.3 ท้นตบุคลากร	22

4.1.4	อาสาสมัครสาธารณสุขประจำหมู่บ้าน (อสม.)	22
	บรรณานุกรม	23

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

ในปัจจุบันปัญหาด้านสุขภาพของประชากรมีแนวโน้มจะสูงขึ้นเรื่อย ๆ ประกอบกับการเข้าสู่สังคมสูงวัยของประชากร ปัญหาสุขภาพจึงเป็นปัญหาที่สำคัญ ซึ่งส่งผลกระทบต่อชีวิตของประชากรโดยส่วนมาก มะเร็งช่องปากเป็นมะเร็งชนิดหนึ่งที่พบบ่อยในกลุ่มประชากรที่มีอายุตั้งแต่ 40 ปีขึ้นไป ที่มีประวัติด้านการสูบบุหรี่ และ/หรือดื่มแอลกอฮอล์ และเคี้ยวหมาก ซึ่งการตรวจสอบรอยโรคในระยะแรกอาจทำได้ยากโดยทั่วไปและหากปล่อยเป็นระยะเวลานานเกินไปอาจทำให้รอยโรคลุกลามเป็นมะเร็งได้ในที่สุด คณะผู้จัดทำมีความสนใจ ในเรื่องนี้ จึงได้จัดทำดิจิทัลแพลตฟอร์มเพื่อรองรับระบบปัญญาประดิษฐ์ (AI) เพื่อใช้ในการตรวจสอบและคัดกรองรอยโรคก่อนมะเร็งช่องปากและมะเร็งช่องปาก ที่ใช้ร่วมกับการประเมินจากทันตแพทย์ผู้เชี่ยวชาญ

คณะผู้จัดทำจึงได้นำเสนอการพัฒนาเว็บแอปพลิเคชัน ซึ่งเป็นแพลตฟอร์มดิจิทัลสำหรับรองรับระบบปัญญาประดิษฐ์ (AI) เพื่อตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและมะเร็งช่องปาก (Digital Platform for Detecting and Analyzing Oral Potentially Malignant Disorders and Oral Cancer) โดยกลุ่มผู้ใช้งานของดิจิทัลแพลตฟอร์มนี้จะเป็นทันตแพทย์ทั่วประเทศและประชากรทั่วไปที่มีความสนใจในการนำดิจิทัลแพลตฟอร์มนี้ไปใช้

คณะผู้จัดทำหวังว่า ดิจิทัลแพลตฟอร์มนี้จะส่งผลให้ทันตแพทย์ทั่วประเทศสามารถตรวจหามะเร็งช่องปากได้อย่างรวดเร็ว และเป็นเครื่องมือหนึ่งที่จะช่วยแก้ไขปัญหาด้านสุขภาพของประชากรโดยเฉพาะมะเร็งช่องปากได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ของโครงการ

1. พัฒนาเว็บแอปพลิเคชันเพื่อรองรับระบบปัญญาประดิษฐ์ (AI)
2. พัฒนาเว็บแอปพลิเคชันเพื่อตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและมะเร็งช่องปาก

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตด้านฮาร์ดแวร์

โครงการนี้ต้องการฮาร์ดแวร์ต่อไปนี้ จึงจะสามารถใช้งานได้อย่างมีประสิทธิภาพ

- คอมพิวเตอร์ส่วนบุคคลหรือโทรศัพท์มือถือที่สามารถใช้งานเว็บเบราว์เซอร์ได้

1.3.2 ขอบเขตด้านซอฟต์แวร์

โครงการนี้ต้องการซอฟต์แวร์ต่อไปนี้ จึงจะสามารถใช้งานได้อย่างมีประสิทธิภาพ

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

1.4 ประโยชน์ที่ได้รับ

ผู้ใช้งาน

- สามารถใช้งานเว็บแอปพลิเคชันเพื่อตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและมะเร็งช่องปากได้
- สามารถเข้าถึงการรักษาทางการแพทย์ได้อย่างรวดเร็ว หลังจาก que ผู้ใช้งานได้รับการตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและมะเร็งช่องปากโดยเว็บแอปพลิเคชัน

ผู้พัฒนา

- ได้รับความรู้และความเข้าใจในการพัฒนาเว็บแอปพลิเคชันเพื่อรองรับระบบปัญญาประดิษฐ์ (AI)
- ได้ฝึกทักษะในการพัฒนาเว็บแอปพลิเคชันเพื่อรองรับระบบปัญญาประดิษฐ์ (AI)
- ได้ฝึกทักษะในการทำงานเป็นทีมและทักษะในการวิเคราะห์และแก้ไขปัญหาที่อาจเกิดขึ้นในการพัฒนา

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านฮาร์ดแวร์

1.5.2 เทคโนโลยีด้านซอฟต์แวร์

- ภาษาโปรแกรมมิ่ง: JavaScript, Python, HTML, CSS
- ฐานข้อมูล: MySQL
- เครื่องมือและเทคโนโลยี: NextJS, Tailwind CSS, Git, GitHub, Google Cloud Platform

1.6 แผนการดำเนินงาน

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

1.7 บทบาทและความรับผิดชอบ

มีหน้าที่และความรับผิดชอบ ดังนี้

นายญาณาริพ ภู่ว่าง รหัสนักศึกษา 630612097 รับผิดชอบในการศึกษาค้นคว้าเทคโนโลยีที่เกี่ยวข้อง, ออกแบบโครงสร้างของเว็บแอปพลิเคชันและพัฒนาเว็บแอปพลิเคชัน

นายปณัฏฐ์ กันทรพิทย์ รหัสนักศึกษา 630612105 รับผิดชอบในการศึกษาค้นคว้าเทคโนโลยีที่เกี่ยวข้อง, ออกแบบโครงสร้างของเว็บแอปพลิเคชันและพัฒนาเว็บแอปพลิเคชัน

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 อาจจะได้รับคำสั่งจาก 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 จะส่งข้อมูลนั้นมายัง 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 ระบบบัญชีภายในของคุณต้องแบ่งปันข้อมูลกับระบบธนาคารของลูกค้าเพื่อออกใบแจ้งหนี้และสื่อสารกับแอปพลิเคชันบันทึกเวลาปฏิบัติงานภายในโดยอัตโนมัติ 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 สำหรับการสร้างเว็บเพจ โดยใช้ภาษา HTML สามารถทำโดยใช้โปรแกรม 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 Tailwind CSS

Tailwind CSS [7] คือ CSS [5] Framework ตัวหนึ่งที่มีรูปแบบการทำงานแบบ Utility-First โดย Utility คือ Class Selector ตัวหนึ่ง ที่เมื่อใช้งานก็เพียงเรียกใช้ Utility ต่างๆมาประกอบกันให้ได้การแสดงผลตามที่เรต้องการ ซึ่งจะมีความต่างกับ CSS Framework อื่นๆที่มักจะกำหนด Class Selector ให้เฉพาะเจาะจงกับรูปแบบการแสดงผลของ Element นั้น ๆ ไปเลย



รูปที่ 2.2: Tailwind CSS



รูปที่ 2.3: Next.js

2.2.5 Next.js

Next.js [8] เป็น open-source React framework ซึ่งต่างจาก React ตรงที่ Next.js เป็นการใช้ server side rendering และยังสามารถทำเว็บไซต์ได้ทั้งแบบ static และ dynamic ซึ่งข้อดีของการเป็น Server Side Rendering คือ ช่วยในเรื่อง SEO หรือ search engine optimization เพราะถ้าทำการ inspect เว็บไซต์ที่สร้างโดย Next.js จะเห็นว่า source จะเป็น html ซะส่วนใหญ่ ซึ่งทำให้ SEO คืบผ่าน source เพื่อให้ได้ข้อมูลและจัดหมวดหมู่ได้ง่ายกว่า React ที่เป็น JavaScript มากกว่า ทำให้ Next.js เป็นที่นิยมในหลาย ๆ บริษัท นอกจากนี้ ข้อดีก็คือ render ได้เร็วกว่า React เพราะ Next.js มีสิ่งที่เรียกว่า get static path ซึ่งการสร้าง path แบบ static แบบเว็บไซต์ html โดยไม่ต้องทำการเชื่อมต่อกับ back-end เพื่อให้ได้ data ยิ่งไปกว่านั้น Next.js สามารถรวมเข้ากับ back-end ได้ง่ายๆ เพราะ Next.js มีสิ่งที่เรียกว่า API routes ในการรับส่ง request ใน folder ของ page จะมีอีก folder ที่เรียกว่า api ที่ถูกปฏิบัติเป็น endpoint แทนที่จะเป็น page ซึ่ง folder api นี้จะเป็นในส่วนของ server-side เท่านั้น ทำให้ไม่ไปเพิ่ม size ของ Client Side

2.2.6 MySQL

MySQL [9] คือ ระบบจัดการฐานข้อมูล หรือ Database Management System (DBMS) แบบข้อมูลเชิงสัมพันธ์ หรือ Relational Database Management System (RDBMS) ซึ่งเป็นระบบฐานข้อมูลที่จัดเก็บรวบรวมข้อมูลในรูปแบบตาราง โดยมีการแบ่งข้อมูลออกเป็นแถว (Row) และในแต่ละแถวแบ่งออกเป็นคอลัมน์ (Column) เพื่อเชื่อมโยงระหว่างข้อมูลในตารางกับข้อมูลในคอลัมน์ที่กำหนด แทนการเก็บข้อมูลที่แยกออกจากกัน โดยไม่มีความเชื่อมโยงกัน ซึ่งประกอบด้วยข้อมูล (Attribute) ที่มีความสัมพันธ์เชื่อมโยงกัน

(Relation) โดยใช้ RDBMS Tools สำหรับการควบคุมและจัดเก็บฐานข้อมูลที่จำเป็น ทำให้นำไปประยุกต์ใช้งานได้ง่าย ช่วยเพิ่มประสิทธิภาพในการทำงานให้มีความยืดหยุ่นและรวดเร็วได้มากยิ่งขึ้น รวมถึงเชื่อมโยงข้อมูล ที่จัดแบ่งกลุ่มข้อมูลแต่ละประเภทได้ตามต้องการ จึงทำให้ MySQL เป็นโปรแกรมระบบจัดฐานข้อมูล ที่ได้รับความนิยมสูง

2.2.7 JSON

JSON [10] ย่อมาจาก (JavaScript Object Notation) เป็นมาตรฐานในการแลกเปลี่ยนข้อมูล (Data Interchange Format) ที่ได้รับความนิยมแทบจะสูงที่สุดในปัจจุบัน ก่อกำเนิดขึ้นในช่วงต้นยุค 2000 ซึ่ง JSON เป็นที่นิยมโดยเฉพาะในงานด้านการทำ APIs ซึ่งเหล่า developers ทุกคนคงรู้จักและคุ้นเคยกันเป็นอย่างดี แม้ว่าจะมีรูปแบบข้อมูลอื่น ๆ อีกมากมายเช่น XML, CSV, YAML, etc เป็นต้น

จุดเด่นของ JSON

- อ่านทำความเข้าใจได้ง่าย
- มีความเบา (lightweight)
- มีความเป็นมาตรฐานสูง และเป็นที่นิยมสูง
- มีความเร็วในการ access ข้อมูลที่สูง เพราะไม่ได้มีโครงสร้างที่ซับซ้อนเหมือนเช่น XML เป็นต้น

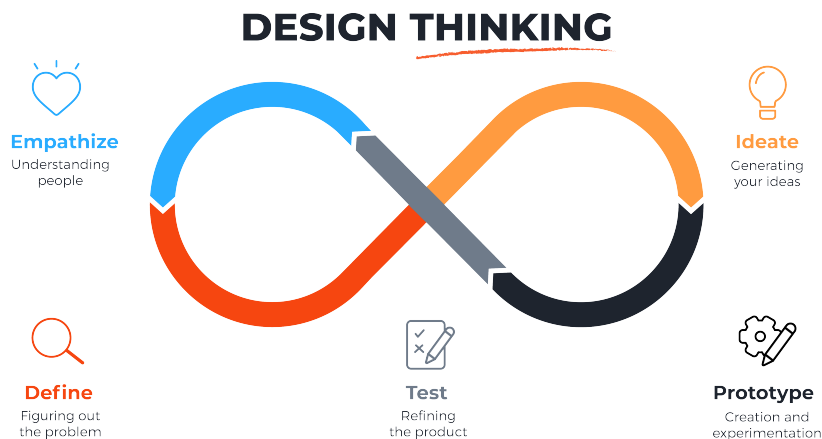
2.3 ด้าน User Interface

ในส่วนนี้จะอธิบายถึงการออกแบบ User Interface ของเว็บแอปพลิเคชัน

2.3.1 Design Thinking

กระบวนการออกแบบ design thinking นั้นมีหลากหลายรูปแบบ ทั้งรูปแบบ 3 ชั้น ไปจนถึง 7 ชั้น ทุกรูปแบบมีความคล้ายคลึงมากที่สุด และใช้หลักการเดียวกันที่อ้างอิงจาก Herbert Simon ผู้ชนะรางวัลโนเบล ในสาขา 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
2. Define หรือ กำหนดปัญหาให้ชัดเจน คือ การเข้าใจความต้องการ ปัญหา และวิเคราะห์ข้อมูลเชิงลึก เพื่อคัดกรองหาปัญหาที่แท้จริง กำหนดหรือบ่งชี้ปัญหาอย่างชัดเจน เพื่อที่จะเป็นแนวทางในการปฏิบัติ และมีทิศทางในการแก้ไขปัญหาอย่างชัดเจน
3. Ideate หรือ ระดมความคิด คือ การนำเสนอแนวคิดต่าง ๆ ร่วมกัน ถึงวิธีการแก้ไขปัญหา อย่างไม่มีกรอบจำกัด การระดมความคิดควรมีมุมมองหลากหลาย และมีหลากหลายแนวทางให้ได้มากที่สุด เพื่อ



รูปที่ 2.4: กระบวนการออกแบบ Design Thinking

ให้มีฐานข้อมูลในการนำไปวิเคราะห์และสรุปผล 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 มาใช้ในการพัฒนาเว็บแอปพลิเคชันทั้งด้านของ front-end ซึ่งจะแสดงผลของเว็บไซต์ และ back-end ที่จะจัดการการทำงานต่าง ๆ รวมถึงการเชื่อมต่อกับฐานข้อมูล
- 261361 Software Engineering การใช้กระบวนการทางวิศวกรรมในการดูแลการผลิต ตั้งแต่การเริ่มเก็บความต้องการ การตั้งเป้าหมายของระบบ การออกแบบ กระบวนการพัฒนา การตรวจสอบ การประเมินผลและทดสอบระบบ
- 261346 Database Systems การใช้งานฐานข้อมูล โดยใช้ MySQL ในการจัดการฐานข้อมูล รวมถึงการเชื่อมต่อกับฐานข้อมูล และการเขียนคำสั่ง SQL ในการดึงข้อมูลจากฐานข้อมูลและการจัดการข้อมูลในฐานข้อมูล

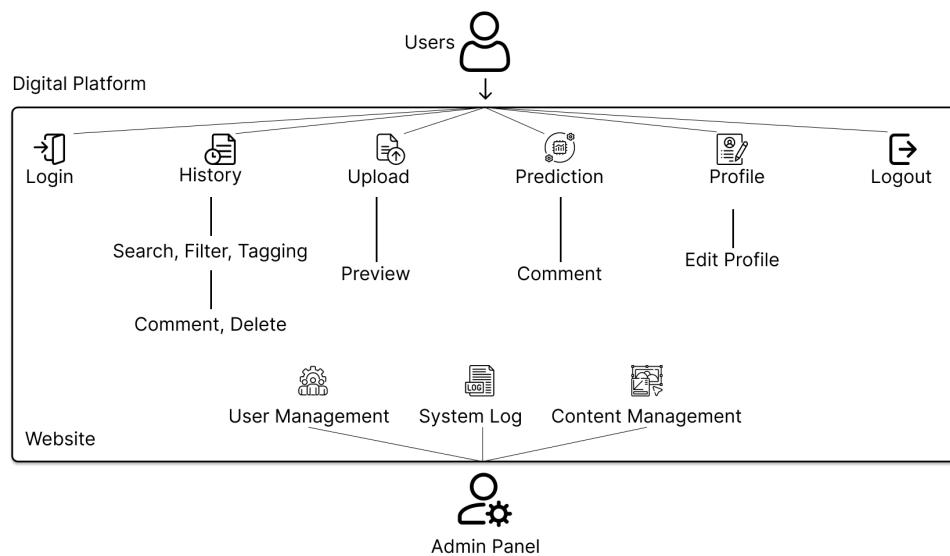
บทที่ 3

โครงสร้างของโครงการ

ในบทนี้จะกล่าวถึงหลักการ และการออกแบบระบบ

3.1 หลักการทำงานของระบบ

3.1.1 ภาพรวมของระบบ (System Overview)



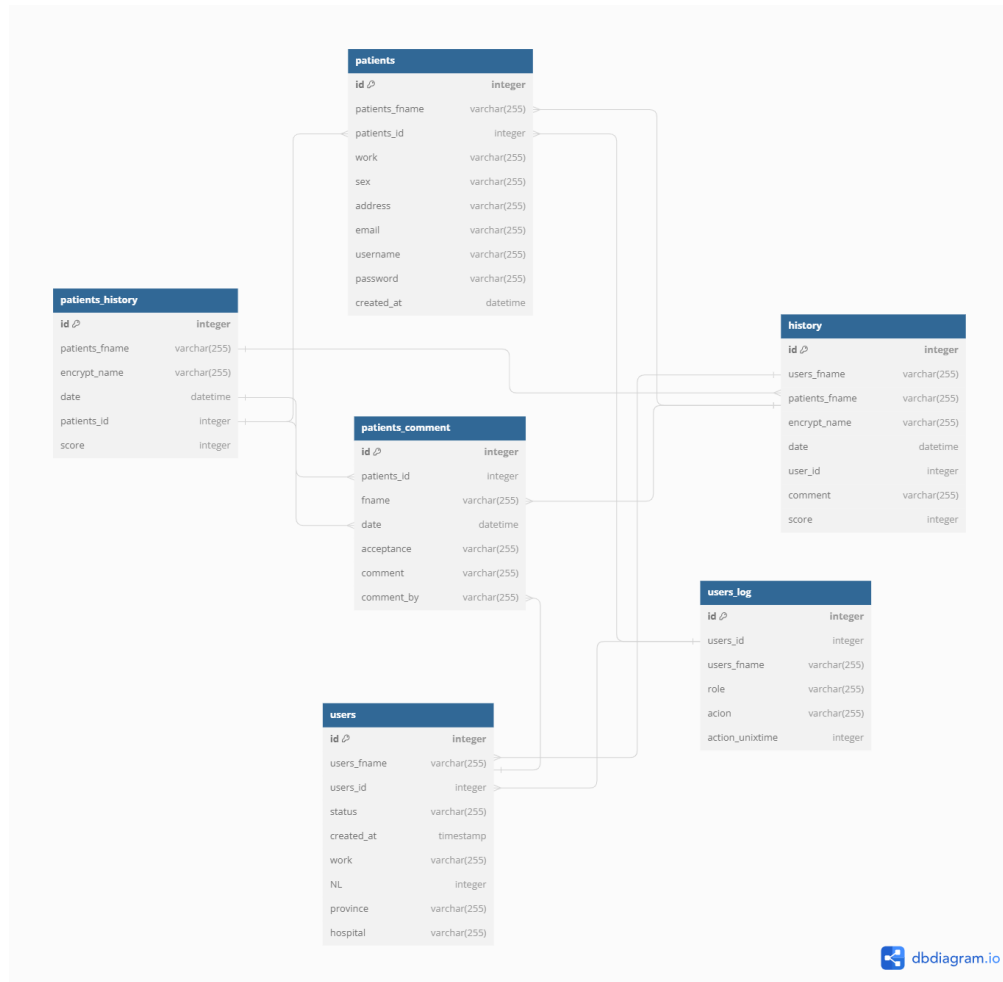
รูปที่ 3.1: System Overview

ภาพรวมการทำงานของระบบนี้ จะมีส่วนการทำงานหลัก ๆ ดังนี้

- ระบบการอัปเดตและการทำนายรอยโรคในช่องปากโดย AI
- ระบบประวัติการอัปเดตและการทำนายรอยโรคในช่องปากสำหรับแต่ละผู้ใช้งาน
- ระบบการตรวจรอยโรคสำหรับทันตแพทย์ (Annotation)
- ระบบการแพทย์ทางไกล (Telemedicine) สำหรับให้ความคิดเห็นระหว่างทันตแพทย์ผู้เชี่ยวชาญกับทันตแพทย์ทั่วไป, อาสาสมัครสาธารณสุขประจำหมู่บ้าน, ทันตบุคลากรและผู้ใช้ทั่วไป
- ระบบจัดการผู้ใช้งาน (User Management) และระบบติดตามผู้ใช้งาน (User Tracking)
- ระบบจัดการข้อมูล (Data Management)

3.1.2 โครงสร้างฐานข้อมูล (Database Schema)

จากรูปที่ 3.2 จะเป็นโครงสร้างของฐานข้อมูลที่ใช้ในระบบ โดยมีรายละเอียดดังนี้



รูปที่ 3.2: Database Schema

Database diagram ของระบบของเรา ได้ทำการแบ่งผู้ใช้งานเป็น 2 ส่วนหลัก ๆ คือ **patients** และ **users** โดย **patients** คือ คนไข้ทั่วไป และ **users** คือกลุ่มทันตแพทย์ พนักงานสาธารณสุขและผู้ดูแลระบบ รวมถึงอสม.

ตาราง **comment** ระบบจะให้เฉพาะทันตแพทย์ที่มีใบ **NL** ในการ **comment** ข้อมูลเท่านั้น โดยสามารถบอกได้ว่าเห็นด้วยหรือไม่เห็นด้วย ด้วยเหตุผลอย่างไร

ตาราง **users_log** เป็น table ที่สามารถดูได้เฉพาะ **user** ที่เป็นผู้ดูแลระบบเท่านั้น โดยจะบอกเกี่ยวกับ **action** ของผู้ใช้งานทุกคนว่า ได้มี **interact** กับระบบอย่างไรบ้าง โดยที่ **action** จะมี **login**, **logout**, **delete** รวมถึง **action** ที่สำคัญต่าง ๆ ไม่ว่าจะเป็น **delete user**, การ **upload image** โดยเก็บเป็น **unixtime** เพื่อให้ง่ายต่อการ **parse**

การเข้ารหัสข้อมูล (**Encryption**) เนื่องจากระบบของเรามีข้อมูลส่วนตัวมากมายที่เก็บเอาไว้ เราจึงต้องทำการเข้ารหัสข้อมูลตาม **requirement** ที่เราได้รับมา โดยจะมี **file** รูปภาพที่เป็นความลับของคนไข้และทำการเข้ารหัส **password** เพื่อความปลอดภัย

3.2 ส่วนเชื่อมต่อระหว่างผู้ใช้งานกับระบบ (User Interface)

3.2.1 ผู้ใช้งาน (User)

หน้าเข้าสู่ระบบ (Login Page)

AIDOC
Lorem ipsum dolor

ลงทะเบียน

Artificial Intelligence Detecting Oral Cancer

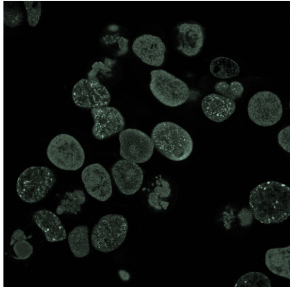
ระบบการวิเคราะห์โรคด้วย AI




ชื่อผู้ใช้งาน

รหัสผ่าน

☐ ลืมรหัสผ่านของคุณ [คลิกที่นี่](#)

เข้าสู่ระบบ





AI V.1.10
AI ในการประมวลผลภาพ

WEB V.1.10
Website ในการส่งข้อมูลภาพ

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 nisi feugiat at tempus sodales amet vel netus at. Enim et sodales cursus metus mauris ullamcorper. Ipsum id ultricies laculis vel sed. Eu non dolor occumsan quisque faucibus blandit morbi praesent.

#2

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



ที่ผู้
Lorem ipsum dolor sit amet consectetur adipiscing elit.

ติดต่อ
Lorem ipsum dolor sit amet consectetur adipiscing elit.

รูปที่ 3.3: Login Page

12

หน้าลงทะเบียน (Register Page)

AIDOC

Lorem ipsum dolor

ลงทะเบียน

ข้อมูลทั่วไปของผู้ใช้งาน

ชื่อ - นามสกุล

ชื่อ - นามสกุล

ตำแหน่ง

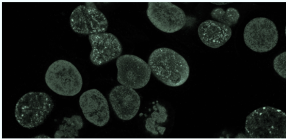
ตำแหน่ง

สถานที่ปฏิบัติงาน

สถานที่ปฏิบัติงาน

โรงพยาบาล

โรงพยาบาล



ข้อมูลติดต่อของผู้ใช้งาน

อีเมล

อีเมล

จังหวัด

จังหวัด

อำเภอ

อำเภอ

ตำบล

ตำบล

เขต/ปริมณฑล

เขต/ปริมณฑล

ชื่อผู้ใช้งาน

ชื่อผู้ใช้งาน

เบอร์โทรศัพท์

เบอร์โทรศัพท์

รหัสผ่าน

รหัสผ่าน

ยืนยันรหัสผ่าน

ยืนยันรหัสผ่าน

ลงทะเบียน

รูปที่ 3.4: Register Page

หน้าประวัติการอัปโหลดรูปภาพ (History Page)

AIDOC

Lorem ipsum dolor

ข้อมูลผู้ใช้งาน

ประวัติ

ตัวอย่างการถ่ายภาพรังสี

เกี่ยวกับ

ผู้ใช้งาน: admin@aidoc.com

สวัสดีคุณนาย, คุณแพทย์หญิง นงุวรรณ เรียบเรียง

คลิกเพื่อดูภาพตัวอย่างของภาพรังสีที่คุณอัปโหลด

ภาพถ่ายของภาพทั้งหมด

5

Folder 1

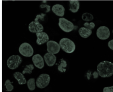
2

Folder 2

3

ภาพถ่ายของปากท้อง

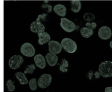
ค้นหาภาพ



ชื่อไฟล์: concert1

วันที่: 25/7/2023 : 15:30 น.

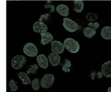
ภาพรังสี: concert1 AI: CPMD



ชื่อไฟล์: concert1

วันที่: 25/7/2023 : 15:30 น.

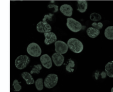
ภาพรังสี: concert1 AI: CPMD



ชื่อไฟล์: concert1

วันที่: 25/7/2023 : 15:30 น.

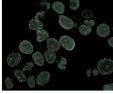
ภาพรังสี: concert1 AI: CPMD



ชื่อไฟล์: concert1

วันที่: 25/7/2023 : 15:30 น.

ภาพรังสี: concert1 AI: CPMD



ชื่อไฟล์: concert1

วันที่: 25/7/2023 : 15:30 น.

ภาพรังสี: concert1 AI: CPMD

หน้าแรก

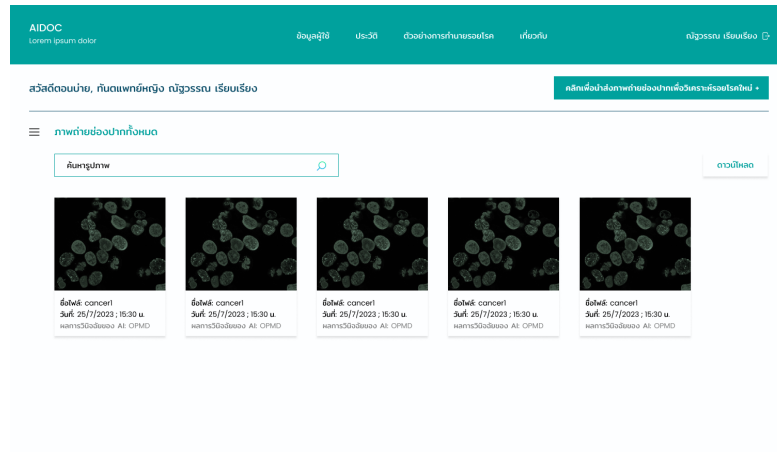
ค้นหา

มีทั้งหมด 1 จาก 1

ถัดไป

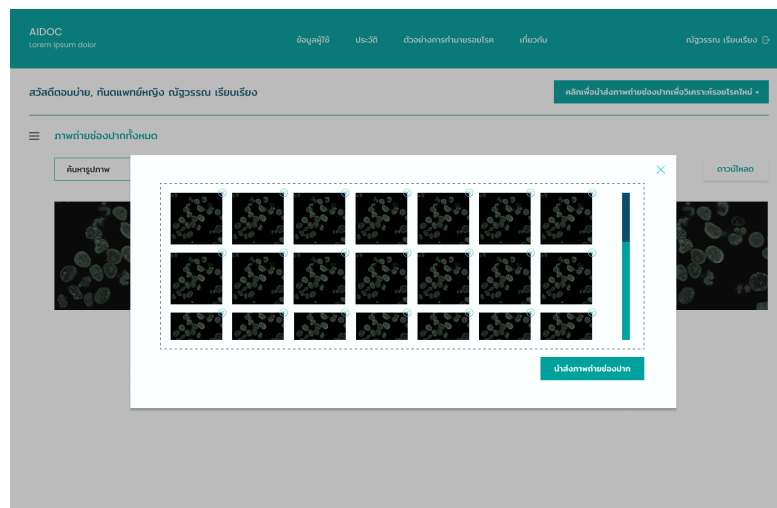
หน้าถัดไป

รูปที่ 3.5: History Page

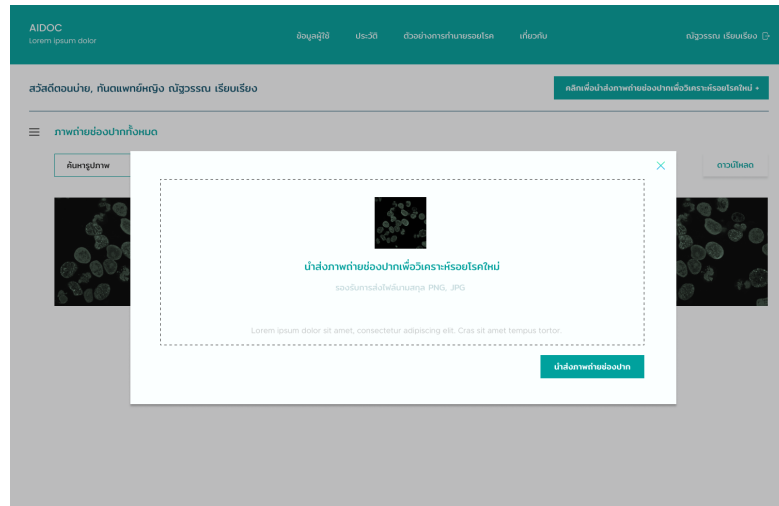


รูปที่ 3.6: History Page

หน้าอัปโหลดรูปภาพ (Upload Page)

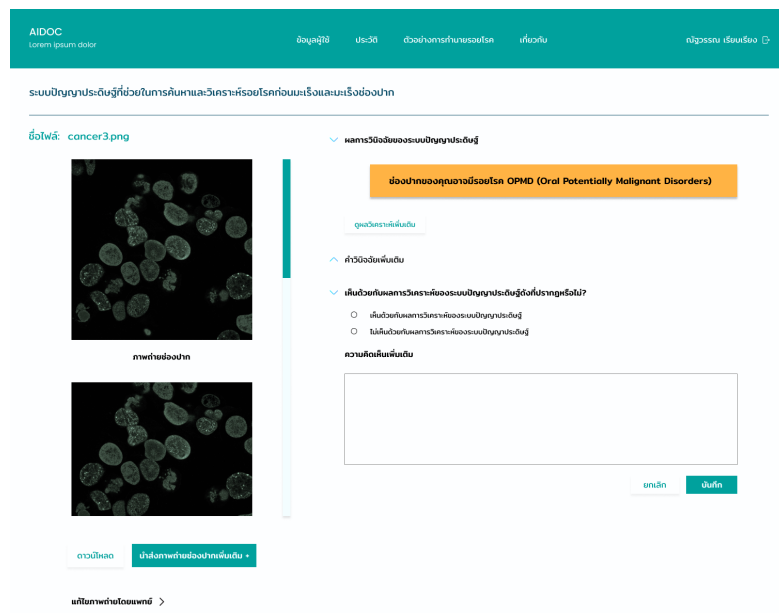


รูปที่ 3.7: Upload Page



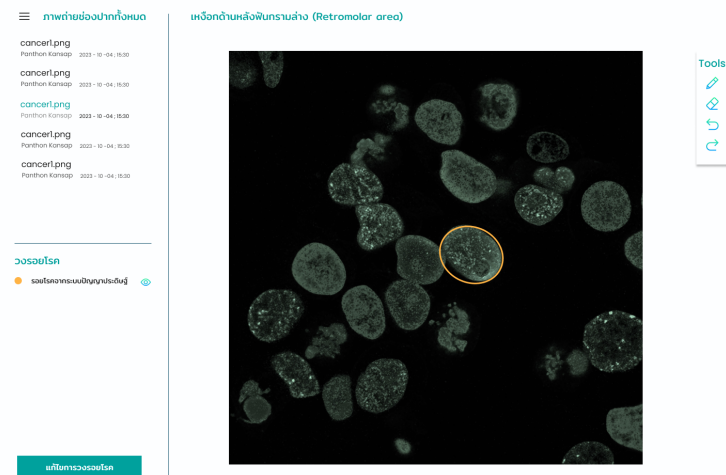
รูปที่ 3.8: Upload Page

หน้าแสดงผลการทำนายโรคในช่องปาก (Result Page)



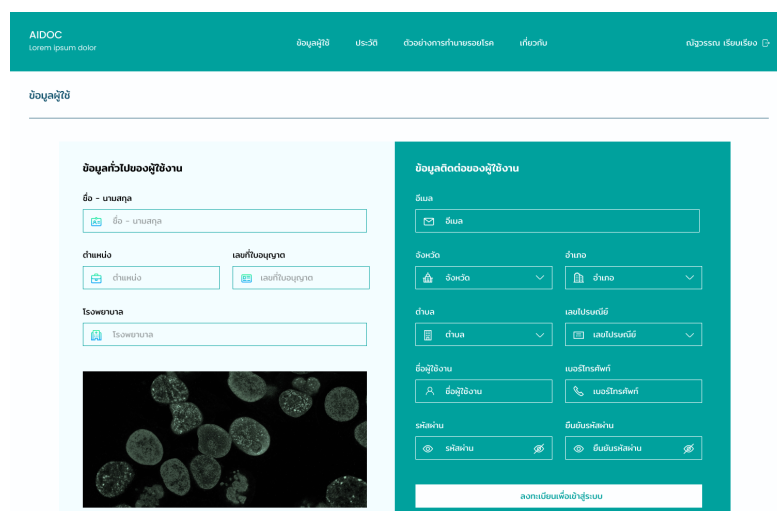
รูปที่ 3.9: Result Page

หน้าสำหรับวางรอยโรคในช่องปาก (Annotation Page)



รูปที่ 3.10: Annotation Page

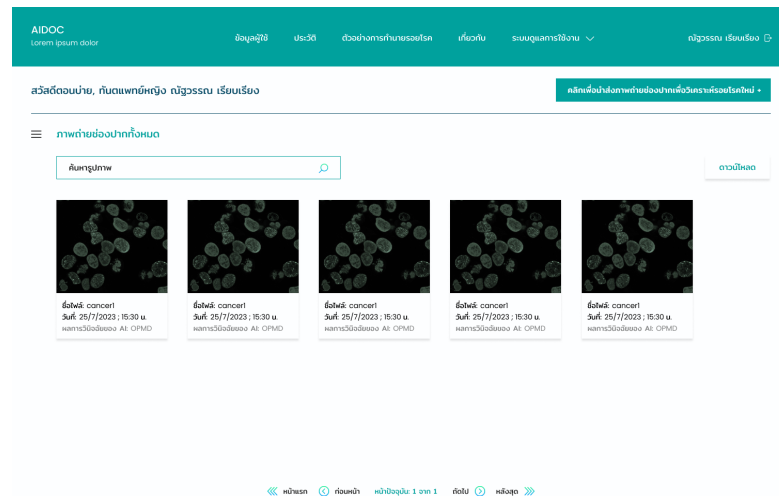
หน้าแก้ไขข้อมูลส่วนตัว (Edit Profile Page)



รูปที่ 3.11: Edit Profile Page

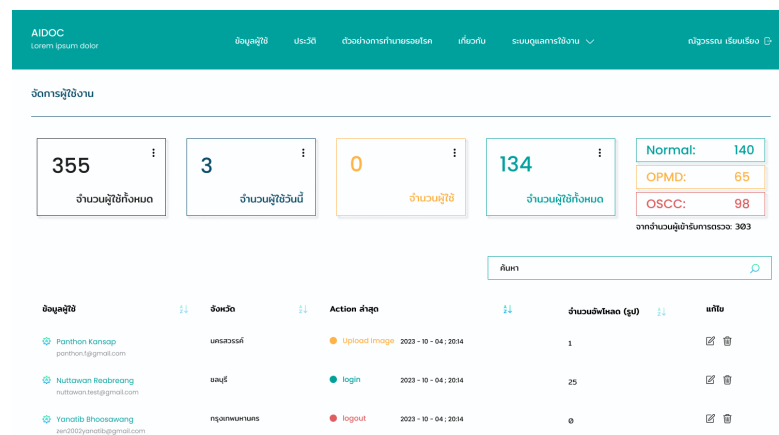
3.2.2 ผู้ดูแลระบบ (Admin)

หน้าอัปโหลดรูปภาพทั้งหมด (Upload Page)



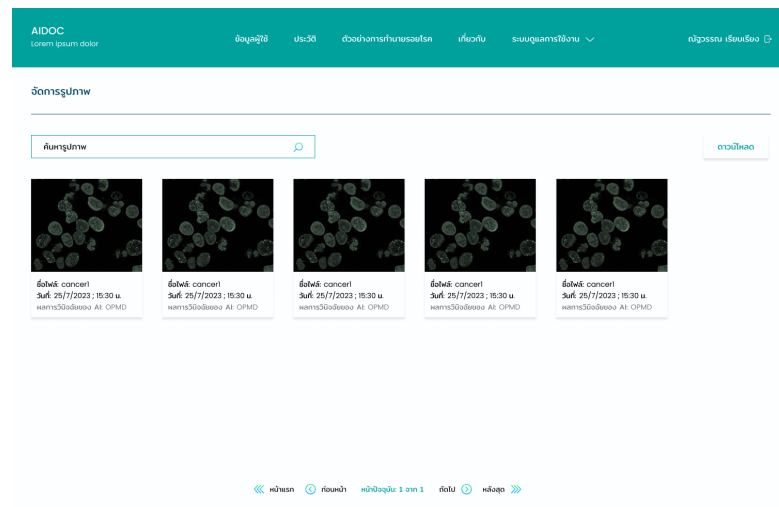
รูปที่ 3.12: Upload Page

หน้าจัดการผู้ใช้งาน (User Management Page)



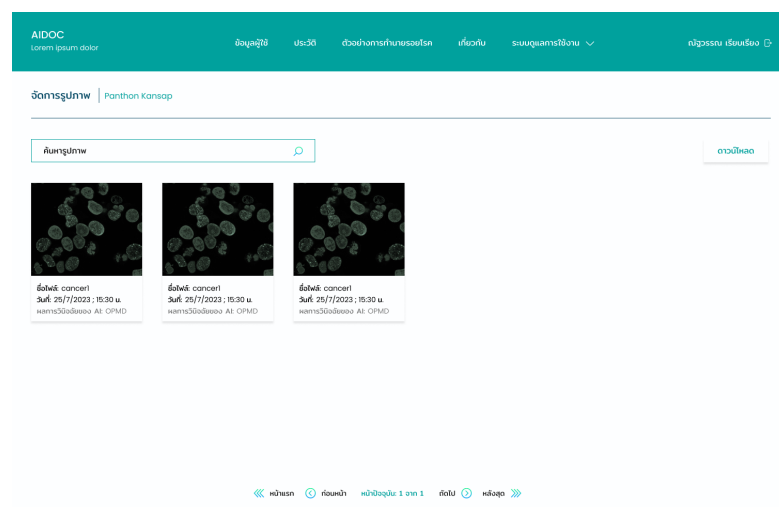
รูปที่ 3.13: User Management Page

หน้าจัดการข้อมูล (Data Management Page)



รูปที่ 3.14: Data Management Page

หน้าจัดการข้อมูล (Data Management Page) โดยแยกตามผู้ใช้งาน




รูปที่ 3.15: Data Management Page

หน้าจัดการผู้ใช้งาน (User Management Page) โดยแยกตามผู้ใช้งาน

ข้อมูลผู้ใช้


ข้อมูลทั่วไปของผู้ใช้งาน

ชื่อ - นามสกุล




ชื่อ - นามสกุล

ตำแหน่ง




ตำแหน่ง

สถานที่ในองค์กร



สถานที่ในองค์กร

โทรศัพท์มือถือ

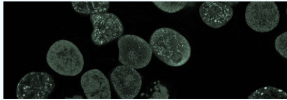


โทรศัพท์มือถือ

บทบาท


User

Admin




ข้อมูลติดต่อของผู้ใช้งาน

อีเมล




อีเมล

จังหวัด




จังหวัด

อำเภอ




อำเภอ

ตำบล




ตำบล

เลขที่ประตูรั้ว




เลขที่ประตูรั้ว

ชื่อผู้ใช้งาน




ชื่อผู้ใช้งาน

เบอร์โทรศัพท์




เบอร์โทรศัพท์

รหัสผ่าน



รหัสผ่าน

ยืนยันรหัสผ่าน



ยืนยันรหัสผ่าน

บันทึกการเปลี่ยนแปลง

รูปที่ 3.16: Data Management Page

บทที่ 4

การประเมินระบบ

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 ในการทดสอบกับผู้ใช้งานทั่วไป ควรเน้นการประเมินปัจจัยต่างๆ เช่น

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

ตัวอย่างวิธีการทดสอบกับผู้ใช้งานทั่วไป 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 ทันทบุคลากร

ทันทบุคลากรมีความต้องการใช้งานระบบที่อำนวยความสะดวกในการทำงาน เช่น การดูประวัติการตรวจคัดกรองมะเร็งช่องปาก การบันทึกข้อมูล การสรุปผลการตรวจคัดกรองมะเร็งช่องปาก ดังนั้น ในการทดสอบกับทันทบุคลากร ควรเน้นการประเมินปัจจัยต่างๆ เช่น

- ความสะดวกในการใช้งาน: Ease of use
- ประโยชน์: Benefits
ตัวอย่างวิธีการทดสอบกับทันทบุคลากร ได้แก่
- ให้ทันทบุคลากรทดสอบระบบและรวบรวมข้อมูลเกี่ยวกับประสบการณ์การใช้งาน เช่น ระยะเวลาในการดำเนินการแต่ละขั้นตอน ความสะดวกในการใช้งาน เป็นต้น
- ให้ทันทบุคลากรประเมินประโยชน์ที่ได้รับจากระบบ 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 ในการทดสอบระบบกับผู้ใช้งานทั่วไป, ทันตแพทย์, ทันทบุคลากรและอสม. จะพิจารณาจากปัจจัยต่าง ๆ เช่น วัตถุประสงค์ของการประเมิน ขอบเขตของการประเมิน ความพร้อมของระบบ เป็นต้น เพื่อให้ได้ผลการประเมินที่มีประสิทธิภาพ

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 เอกซ์เอ็มแอล 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-ทั้งที/>
- [7] D. Image. (2022) Tailwind css คืออะไร. เข้าชมล่าสุดเมื่อ: 3 ตุลาคม 2566. [Online]. Available: <https://medium.com/@dektaiimage/ในบทความนี้เราจะมาพูดถึง-css-framework-ตัวหนึ่งที่มาแรงเป็นที่นิยมใช้กัน-ที่มีชื่อว่า-tailwind-css-132bf1ed6790>
- [8] L. Frevation Co. (2021) Next.js. เข้าชมล่าสุดเมื่อ: 3 ตุลาคม 2566. [Online]. Available: <https://www.frevation.com/blog/web-development/next-js/>
- [9] openlandscape. (2023) Mysql คือ อะไร ? โปรแกรมจัดการฐานข้อมูล open source ยอดนิยม ! เข้าชมล่าสุดเมื่อ: 3 ตุลาคม 2566. [Online]. Available: <https://blog.openlandscape.cloud/mysql>
- [10] D. Team. (2023) Json คืออะไร? พร้อมวิธีการใช้งาน. เข้าชมล่าสุดเมื่อ: 3 ตุลาคม 2566. [Online]. Available: <https://devhub.in.th/blog/what-is-json>