

โครงการเลขที่ วศ.คพ. P019-2/2566

เรื่อง

แพลตฟอร์มดิจิทัลสำหรับการตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและ
มะเร็งซ่องปาก

โดย

นายชาญชัย ไชยสลี รหัส 630610726
นายเทพฤทธิ์ สมฤทธิ์ รหัส 630610731

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2566

PROJECT No. CPE P019-2/2566

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

Chanchai Chaisalee 630610726

Tewarad Somrad 630610731

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 : แพลตฟอร์มดิจิทัลสำหรับการตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็ง และมะเร็งซ่องปาก
โดย : นายชาญชัย ไชยสกี รหัส 630610726
นายเทวฤทธิ์ สมฤทธิ์ รหัส 630610731
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : รศ.ดร. ปภิเวช วุฒิสารวัฒนา¹
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2566

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

คณะกรรมการสอบโครงการ

หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์

(รศ.ดร. สันติ พิทักษ์กิจนุกุร)

ประธานกรรมการ

(รศ.ดร. ปภิเวช วุฒิสารวัฒนา)

กรรมการ

(ผศ. โดม โพธิ์กานนท์)

กรรมการ

(รศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล)

หัวข้อโครงการ : แพลตฟอร์มดิจิทัลสำหรับการตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็ง และมะเร็งช่องปาก
โดย : นายชาญชัย ไชยสกุล รหัส 630610726
นายเทวฤทธิ์ สมฤทธิ์ รหัส 630610731
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : รศ.ดร. ปภิเวช วุฒิสารวัฒนา¹
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2566

บทคัดย่อ

แพลตฟอร์มดิจิทัลสำหรับการตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็ง และมะเร็งช่องปาก เป็น■
เว็บแอปพลิเคชันสำหรับรองรับระบบปัญญาประดิษฐ์ (AI) เพื่อตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรค
ก่อนมะเร็ง และมะเร็งช่องปาก (Digital Platform for Detecting and Analyzing Oral Potentially
Malignant Disorders and Oral Cancer) เป็นต้นได้ด้วยตนเอง กลุ่มผู้ใช้งานของดิจิทัลแพลตฟอร์มนี้จะ
เป็นทันตแพทย์และประชาชนทั่วไป โดยการยืนยันผลการตรวจของระบบปัญญาประดิษฐ์จะถูกยืนยันผลจาก
ทันตแพทย์ที่เข้าร่วมโครงการ เมื่อตรวจสอบพบว่ามีรอยโรคจริงก็ดำเนินการรักษาในขั้นต่อไป

Project Title : Digital Platform for Detecting and Analyzing Oral Potentially Malignant Disorders and Oral Cancer

Name : Chanchai Chaisalee 630610726
Tewarad Somrad 630610731

Department : Computer Engineering

Project Advisor : Assoc. Prof. Patiwet Wuttisarnwattana, Ph.D.

Degree : Bachelor of Engineering

Program : Computer Engineering

Academic Year : 2023

ABSTRACT

Digital Platform for Detecting and Analyzing Oral Potentially Malignant Disorders and Oral Cancer is web application integrated with AI for self detecting oral potentially malignant disorders and oral cancer. User group are dentist and general public. AI analyzed result will confirmed by dentist participating in the project. If result is verified to be disorders, proceed with treatment in the next step.

กิตติกรรมประกาศ

โครงการนี้จะไม่สำเร็จลุล่วงลงได้ถ้าหากไม่ได้รับความกรุณาจาก รศ.ดร.ปภิเวช วุฒิสารวัฒนา อาจารย์ที่ปรึกษา โครงการ ที่ได้สละเวลาส่วนตัวมาให้ความช่วยเหลือแก่โครงการนี้ โดยได้ให้คำเสนอแนะ แนวคิด ซึ่งทางการ หาความรู้ที่จำเป็นในการทำเว็บแอปพลิเคชัน ตลอดจนช่วยตรวจสอบแก้ไขข้อบกพร่องต่างๆ มาโดยตลอด รวมถึง พศ.โดม โพธิกานนท์ และ รศ.ดร. ศันสนีย์ เอื้อพันธุ์วิริยะกุล ที่ให้คำปรึกษา คำแนะนำ จนทำให้ โครงการนี้มีความสมบูรณ์มากที่สุด

ขอบคุณคณชวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ที่ให้สถานที่ในการทำโครงการ ทั้งห้องภาควิชา วิศวกรรมคอมพิวเตอร์ และสถานที่ต่างๆในภาควิชา และยังให้การสนับสนุนทางด้านงบประมาณ อุปกรณ์ ต่างๆ ที่จำเป็นต่อการทำโครงการ

ขอบคุณผู้ช่วยที่ให้การสนับสนุน ที่ให้คำปรึกษา คำแนะนำ และคอยเป็นกำลังใจ ให้ตลอด มา ซึ่งเป็นแรงผลักดันให้แก่ผู้จัดทำมีความตั้งใจและมุ่งมั่นในการทำงาน จนโครงการที่ความสมบูรณ์ มากที่สุด นอกจากนี้ผู้จัดทำขอขอบพระคุณอีกหลายท่านที่ไม่ได้กล่าวถึง ณ ที่นี่ ที่ได้ให้ความช่วยเหลือตลอดมา และสุดท้ายนี้ หากโครงการนี้มีข้อผิดพลาดประการใด ผู้จัดทำขออภัยมา ณ ที่นี่ และพร้อมน้อมรับด้วยความ ยินดี acknowledgment environment.

นายชาญชัย ไชยสลี
นายเทวฤทธิ์ สมฤทธิ์
25 พฤษภาคม 2563

สารบัญ

บทคัดย่อ	๑
Abstract	๒
กิตติกรรมประกาศ	๓
สารบัญ	๔
สารบัญรูป	๕
สารบัญตาราง	๖
1 บทนำ	1
1.1 ที่มาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.3.1 ขอบเขตด้านอาร์ดแวร์	1
1.3.2 ขอบเขตด้านซอฟต์แวร์	1
1.4 ประโยชน์ที่ได้รับ	2
1.5 เทคโนโลยีและเครื่องมือที่ใช้	2
1.5.1 เทคโนโลยีด้านอาร์ดแวร์	2
1.5.2 เทคโนโลยีด้านซอฟต์แวร์	2
1.6 แผนการดำเนินงาน	2
1.7 บทบาทและความรับผิดชอบ	2
1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม	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	6
2.2.3 TypeScript	7
2.2.4 Tailwind CSS	7
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
3 โครงสร้างและขั้นตอนการทำงาน	10
3.1 หลักการทำงานของระบบ	10
3.1.1 ภาพรวมของระบบ (System Overview)	10
3.1.2 โครงสร้างฐานข้อมูล (Database Schema)	10
3.2 ส่วนเชื่อมต่อระหว่างผู้ใช้งานกับระบบ (User Interface)	10
3.2.1 หันตบุคลากรและผู้ใช้ทั่วไป (User)	11
3.2.2 ประชาชนทั่วไป (General public)	15
3.2.3 อาสาสมัครสาธารณสุขประจำหมู่บ้าน (อสม.)	17

4 การทดลองและผลลัพธ์	20
4.1 การทดลองเกี่ยวกับการทำงานของระบบ	20
4.1.1 ผู้ใช้ทั่วไป	20
4.1.2 ทันตแพทย์	20
4.1.3 ทันตบุคลากร	21
4.1.4 อาสาสมัครสาธารณสุขประจำหมู่บ้าน (อสม.)	21
5 บทสรุปและข้อเสนอแนะ	22
5.1 สรุปผล	22
5.2 ปัญหาที่พบและแนวทางการแก้ไข	22
5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ	22

สารบัญรูป

3.1 login page	11
3.2 register page	11
3.3 history page	12
3.4 history page: comment	12
3.5 upload page	12
3.6 preview page	13
3.7 uploaded page	13
3.8 checkup page	13
3.9 checking page	14
3.10 checking page: comment	14
3.11 report page	14
3.12 login page	15
3.13 register page	15
3.14 upload page	16
3.15 preview page	16
3.16 uploaded page	16
3.17 login page	17
3.18 register page	17
3.19 history page	18
3.20 upload page	18
3.21 preview page	18
3.22 uploaded page	19

ສາຮບໍ່ຢູ່ຕາຮາງ

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

ในปัจจุบันปัญหาด้านสุขภาพของประชาชนมีแนวโน้มจะสูงขึ้นเรื่อยๆ ประกอบกับการเข้าสู่สังคมสูงวัยของ ประชาชน ปัญหาสุขภาพจึงเป็นปัญหาที่สำคัญ ซึ่งส่งผลกระทบต่อชีวิตของประชาชนโดยส่วนมาก มะเร็งซึ่ง ปากเป็นมะเร็งชนิดหนึ่งที่พบมากในกลุ่มประชาชนที่มีอายุตั้งแต่ 40 ปี ขึ้นไป ที่มีประวัติด้านการสูบบุหรี่ และ/หรือ ดื่มแอลกอฮอล์ และเคี้ยวหมาก ซึ่งการตรวจสอบรอยโรคในระยะแรกอาจทำได้ยากโดยทั่วไปและหากปล่อย เป็นระยะเวลานานกินไปอาจทำให้รอยโรคลุกลามเป็นมะเร็งได้ในที่สุด

คณะผู้จัดทำมีความสนใจในเรื่องนี้ จึงได้จัดทำดิจิทัลแพลตฟอร์มเพื่อรองรับระบบปัญญาประดิษฐ์ (AI) เพื่อใช้ในการตรวจสอบและคัดกรองรอยโรคก่อนมะเร็งซึ่งปากและมะเร็งซึ่งปาก ที่ใช้ร่วมกับการประเมิน จากทันตแพทย์ผู้เชี่ยวชาญ คณะผู้จัดทำจึงได้นำเสนอการพัฒนาเว็บแอปพลิเคชัน ซึ่งเป็นแพลตฟอร์มดิจิทัล สำหรับรองรับระบบปัญญาประดิษฐ์ (AI) เพื่อตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและมะเร็งซึ่งปาก (Digital Platform for Detecting and Analyzing Oral Potentially Malignant Disorders and Oral Cancer) โดยกลุ่มผู้ใช้งานของดิจิทัลแพลตฟอร์มนี้จะเป็นทันตแพทย์ทั่วประเทศและประชาชนทั่วไปที่มีความสนใจในการนำดิจิทัลแพลตฟอร์มนี้ไปใช้

คณะผู้จัดทำหวังว่า ดิจิทัลแพลตฟอร์มนี้จะส่งผลให้ทันตแพทย์ทั่วประเทศสามารถตรวจหามะเร็งซึ่งปาก ได้อย่างรวดเร็ว และเป็นเครื่องมือหนึ่งที่จะช่วยแก้ไขปัญหาด้านสุขภาพของประชาชนโดยเฉพาะมะเร็งซึ่งปาก ได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ของโครงการ

- พัฒนาเว็บแอปพลิเคชันเพื่อรองรับระบบปัญญาประดิษฐ์(AI)
- พัฒนาเว็บแอปพลิเคชันเพื่อตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและมะเร็งซึ่งปาก

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตด้านฮาร์ดแวร์

โครงการนี้ต้องการฮาร์ดแวร์ต่อไปนี้ จึงจะสามารถใช้งานได้อย่างมีประสิทธิภาพ

- คอมพิวเตอร์ส่วนบุคคลหรือโทรศัพท์มือถือที่สามารถใช้งานเว็บเบราว์เซอร์ได้

1.3.2 ขอบเขตด้านซอฟต์แวร์

โครงการนี้ต้องการซอฟต์แวร์ต่อไปนี้ จึงจะสามารถใช้งานได้อย่างมีประสิทธิภาพ

- สามารถใช้งานเว็บไซต์บนระบบปฏิบัติการทั่วไปได้ เช่น Windows, macOS, Linux, Android, iOS และอื่นๆ

1.4 ประโยชน์ที่ได้รับ

ผู้ใช้งาน

- สามารถใช้งานเว็บแอปพลิเคชันเพื่อตรวจสอบคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและมะเร็งซ่องปากได้

- สามารถเข้าถึงการรักษาทางการแพทย์ได้อย่างรวดเร็ว หลังจากที่ผู้ใช้งานได้รับการตรวจคัดกรองและเฝ้าระวังการเกิดรอยโรคก่อนมะเร็งและมะเร็งซ่องปากโดยเว็บแอปพลิเคชัน

ผู้พัฒนา

- ได้รับความรู้และความเข้าใจในการพัฒนาเว็บแอปพลิเคชันเพื่อรับระบบปัญญาประดิษฐ์(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, Minio

1.6 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	พ.ย. 2566	ธ.ค. 2566	ม.ค. 2567	ก.พ. 2567
ศึกษาค้นคว้าเกี่ยวกับเทคโนโลยีที่เกี่ยวข้อง				
ออกแบบพื้นฐานที่จะเพิ่มในเว็บแอปพลิเคชัน				
พัฒนาฟีเจอร์ตามที่ได้ออกแบบไว้				
ทดสอบและปรับปรุง				

1.7 บทบาทและความรับผิดชอบ

มีหน้าที่และความรับผิดชอบ ดังนี้

นายชาญชัย ไชยสตี รหัสนักศึกษา 630610726 รับผิดชอบในการศึกษาค้นคว้าเทคโนโลยีที่เกี่ยวข้อง, ออกแบบโครงสร้างของเว็บแอปพลิเคชันและพัฒนาเว็บแอปพลิเคชัน

นายเทวฤทธิ์ สมฤทธิ์ รหัสนักศึกษา 630610731 รับผิดชอบในการศึกษาค้นคว้าเทคโนโลยีที่เกี่ยวข้อง, ออกแบบโครงสร้างของเว็บแอปพลิเคชันและพัฒนาเว็บแอปพลิเคชัน

1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

โครงการนี้จัดทำขึ้นเพื่อให้ประชาชนทั่วไปได้มีโอกาสเข้าถึงการรักษาได้มากยิ่งขึ้น เพิ่มโอกาสการรักษาสำเร็จให้สูงขึ้นได้ ถ้าตรวจพบโรคได้เร็วพอ แต่ทั้งนี้ก็ต้องอาศัยความร่วมมือจากหน่วยงานทางการแพทย์ที่นำไปใช้งานจริง

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ด้านโครงสร้างเว็บแอปพลิเคชัน

ในส่วนนี้จะอธิบายถึงโครงสร้างของเว็บแอปพลิเคชันที่ใช้ในการพัฒนา

2.1.1 MVC Architecture

MVC [1] เป็นตัวย่อของคำว่า Model View Controller ใช้เรียกรูปแบบการพัฒนาซอฟต์แวร์ที่มีโครงสร้างซึ่งแบ่งออกมาเป็น 3 ส่วนหลัก ตามตัวย่อของชื่อรูปแบบการพัฒนาซอฟต์แวร์แบบ MVC ถูกนำไปใช้ในขั้นตอนการพัฒนาหลากหลายภาษา เพราะ MVC เป็นเพียงหลักการออกแบบโปรแกรม (Design Pattern) รูปแบบหนึ่งเท่านั้น ซึ่งเป็นที่นิยมมาก ในการนำมาพัฒนาแอปพลิเคชันซอฟต์แวร์แต่ละแพลตฟอร์ม และประยุกต์ใช้ในอีกหลาย ๆ ที่กัน

ส่วนของ Model (M)

model คือส่วนของการเก็บรวบรวมข้อมูล ไม่ว่าข้อมูลนั้น ๆ จะถูกจัดเก็บในรูปแบบใดก็ตาม ในฐานข้อมูลแบบเป็น Object Class หรือที่นิยมเรียกว่า VO (Value Object) หรือเก็บเป็นไฟล์ข้อมูลเลย เมื่อข้อมูลถูกโหลดเข้ามาจากที่ต่าง ๆ และเข้ามายังส่วนของโมเดล ตัวโมเดลจะทำการจัดการตระเตรียมข้อมูลให้ เป็นรูปแบบที่เหมาะสม เพื่อรอการร้องขอข้อมูลจากส่วนของ Controller

ส่วนของ View (V)

view คือส่วนของการแสดงผล หรือส่วนที่จะปฏิสัมพันธ์กับผู้ใช้งาน (User Interface) หน้าที่ของ view 在การเขียนโปรแกรมแบบ MVC คืออยู่รับคำสั่งจากส่วนของ Controller และ End User เริ่มแรกโดยตัววิว อาจจะได้รับคำสั่งจาก Controller ให้แสดงผลหน้า Home และเมื่อผู้ใช้งานหน้าเว็บกดปุ่มสั่งซื้อ View จะส่งข้อมูลไปให้ Controller เพื่อประมวลผลและแสดงบางอย่างจาก Action นั้น

ส่วนของ Controller (C)

controller คือส่วนของการเริ่มทำงาน และรับคำสั่ง โดยที่คำสั่งนั้นจะเกิดขึ้นในส่วนการติดต่อกับผู้ใช้งาน คือ view เมื่อผู้ใช้งานทำการ Interactive กับ UI view จะเกิดเหตุการณ์หรือข้อมูลบางอย่างขึ้น ตัววิวจะส่งข้อมูลนั้น หมายความว่า 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 แอปพลิเคชันทางธุรกิจส่วนใหญ่ต้องสื่อสารกับแอปพลิเคชันภายนอกในอื่นๆ และของบุคคล

ที่สามเพื่อทำงานต่างๆ ตัวอย่างเช่น หากต้องการสร้างสิ่งใดสิ่งหนึ่ง ระบบบัญชีภายในของคุณต้องแบ่งปันข้อมูลกับระบบธนาคารของลูกค้าเพื่อออกใบแจ้งหนี้และสื่อสารกับแอปพลิเคชันบันทึกเวลาปฏิบัติงานภายใน โดยอัตโนมัติ RESTful API ให้การสนับสนุนการแลกเปลี่ยนข้อมูลนี้ เพราะเป็นระบบที่มีมาตรฐานการสื่อสารระหว่างซอฟต์แวร์ที่ปลอดภัย เช่น Java และมีประสิทธิภาพ

API (Application Programming Interface)

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

ซอฟต์แวร์ที่ใช้ API ก็ได้ ตัวอย่างเช่น นักพัฒนาสามารถเขียนโปรแกรมที่เข้าถึงข้อมูลสภาพอากาศจากระบบสภาพอากาศ หรือคุณสามารถเข้าถึงข้อมูลเดียวกันจากเบราว์เซอร์เมื่อคุณเยี่ยมชมเว็บไซต์รายงานสภาพอากาศได้โดยตรง

ทรัพยากร ทรัพยากรคือข้อมูลที่แอปพลิเคชันต่างๆ มอบให้แก่ซอฟต์แวร์ โดยทรัพยากรอาจเป็นรูปภาพ วิดีโอ ข้อความ ตัวเลข หรือข้อมูลประเภทใดก็ได้ ทั้งนี้เครื่องคอมพิวเตอร์ที่มอบทรัพยากรให้แก่ซอฟต์แวร์นั้นเรียกว่าตัวเซิร์ฟเวอร์ องค์กรต่างๆ ใช้ API เพื่อแบ่งปันทรัพยากรและให้บริการเว็บในขณะที่ยังคงดูแลรักษาความปลอดภัย การควบคุม และการรับรองความถูกต้องไปพร้อมกัน นอกจากนี้ API ยังช่วยให้ลูกค้าระบุได้ว่าซอฟต์แวร์สามารถเข้าถึงทรัพยากรภายในที่เฉพาะเจาะจงได้

REST (Representational State Transfer)

REST เป็นสถาปัตยกรรมซอฟต์แวร์ที่กำหนดเงื่อนไขว่า API ควรทำงานอย่างไร โดยแต่แรกเริ่มนั้น มีการสร้าง REST ขึ้นเพื่อเป็นแนวทางในการจัดการการสื่อสารบนเครือข่ายที่ซับซ้อน เช่น อินเทอร์เน็ต คุณสามารถใช้สถาปัตยกรรม REST เพื่อรับการสื่อสารที่มีประสิทธิภาพสูงและเข้าถึงได้ในทุกระดับ คุณยังสามารถใช้และปรับเปลี่ยนสถาปัตยกรรมได้อย่างง่ายดาย โดยนำความสามารถในการมองเห็นและการเคลื่อนย้ายข้อมูลแพลตฟอร์มมาสู่ทุกระบบ 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 ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่าง มีระบบ มีความสม่ำเสมอระหว่างข้อมูลต่าง ๆ ที่ซัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูลเกี่ยวกับสิ่งของสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งาน

และดูแล รักษาป้องกันข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และ โปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้งานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (data base management system) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูล ของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งค่าตามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

ประโยชน์ของฐานข้อมูล

- ลดการเก็บข้อมูลที่ซ้ำซ้อน ข้อมูลบางชุดที่อยู่ในรูปของแฟ้มข้อมูลอาจมีปรากฏอยู่หลาย ๆ แห่ง เพราะมีผู้ใช้ข้อมูลชุดนี้หลายคน เมื่อใช้ระบบฐานข้อมูลแล้วจะช่วยให้ความซ้ำซ้อนของข้อมูลลดน้อยลง
- รักษาความถูกต้องของข้อมูล เนื่องจากฐานข้อมูลมีเพียงฐานข้อมูลเดียว ในกรณีที่มีข้อมูลชุดเดียวกันปรากฏอยู่หลายแห่งในฐานข้อมูล ข้อมูลเหล่านี้จะต้องตรงกัน ถ้ามีการแก้ไขข้อมูลนี้ทุก ๆ แห่งที่ข้อมูลปรากฏอยู่จะแก้ไขให้ถูกต้อง ตามกันหมดโดยอัตโนมัติด้วยระบบจัดการฐานข้อมูล
- การป้องกันและรักษาความปลอดภัยให้กับข้อมูลทำได้อย่างสะดวก การป้องกันและรักษาความปลอดภัยกับข้อมูลระบบฐานข้อมูลจะให้เฉพาะผู้ที่เกี่ยวข้องเท่านั้น ซึ่งก่อให้เกิดความปลอดภัย (security) ของข้อมูลด้วย

2.2 ด้านเทคโนโลยี

ในส่วนนี้จะอธิบายถึงเทคโนโลยีที่ใช้ในการพัฒนาเว็บแอปพลิเคชัน

2.2.1 HTML

HTML [4] ย่อมาจาก Hyper Text Markup Language คือภาษาคอมพิวเตอร์ที่ใช้ในการแสดงผลของเอกสารบน website หรือที่เราระยกันว่าเว็บเพจ 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 ทำการเรียกใช้งานหรือทดสอบการทำงานของเอกสาร HTML จะใช้โปรแกรม web browser เช่น Google Chrome, Microsoft Edge, Mozilla Firefox, Safari และ Opera เป็นต้น

2.2.2 CSS

CSS [5] ย่อมาจาก Cascading Style Sheet มักเรียกโดยย่อว่า "สีต่ำชีต" คือภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS กำหนดกฎเกณฑ์ในการระบุรูปแบบ (หรือ Style) ของเนื้อหาในเอกสาร ยังไง才แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งการกำหนดรูปแบบ หรือ Style นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสาร ไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลลัพธ์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสาร บ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วไป

ทุก หน้าเอกสารภายในเว็บไซต์เดียวกัน โดยกฎเกณฑ์ในการกำหนดรูปแบบ (Style) เอกสาร HTML ถูกเพิ่มเข้า มาครั้งแรกใน 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 จะช่วยให้คุณสามารถสร้าง 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 ต่างๆมาประกอบกันให้ได้การแสดงผลตามที่เราต้องการ ซึ่งจะมีความต่างกับ CSS Framework อื่นที่มักจะกำหนด Class Selector ให้เฉพาะเจาะจงกับรูปแบบการแสดงผลของ Element นั้น ๆ ไปเลย

2.2.5 Next.js

Next.js [8] เป็น open-source React framework ซึ่งต่างจาก React ตรงที่ Next.js เป็นการใช้server 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 ค้นผ่าน 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 โดยไม่ต้องทำการเขียนต่อ กับ 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 นี้จะเป็นในส่วนหนึ่งของ server-side เท่านั้น ทำให้มีไปเพิ่ม size ของ Client Side

2.2.6 MySQL

MySQL [9] คือ ระบบจัดการฐานข้อมูล หรือ Database Management System (DBMS) แบบข้อมูลเชิงสัมพันธ์ หรือ Relational Database Management System (RDBMS) ซึ่งเป็นระบบฐานข้อมูลที่จัดเก็บรวมรวมข้อมูลในรูปแบบตาราง โดยมีการแบ่งข้อมูลออกเป็นแถว (Row) และในแต่ละแถวแบ่งออกเป็น คอลัมน์ (Column) เพื่อเชื่อมโยงระหว่างข้อมูลในตารางกับข้อมูลในคอลัมน์ที่กำหนด แทนการเก็บข้อมูลที่แยกออกจากกัน โดยไม่มีความเชื่อมโยงกัน ซึ่งประกอบด้วยข้อมูล (Attribute) ที่มีความสัมพันธ์เชื่อมโยงกัน (Relation) โดยใช้ RDBMS Tools สำหรับการควบคุมและจัดเก็บฐานข้อมูลที่จำเป็น ทำให้นำไปประยุกต์

ใช้งานได้ง่าย ช่วยเพิ่มประสิทธิภาพในการทำงานให้มีความยืดหยุ่นและรวดเร็วได้มากยิ่งขึ้น รวมถึงเชื่อมโยงข้อมูล ที่จัดแบ่งกลุ่มข้อมูลแต่ละประเภทได้ตามต้องการ จึงทำให้ MySQL เป็นโปรแกรมระบบจัดฐานข้อมูลที่ได้รับความนิยมสูง

2.2.7 JSON

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

จุดเด่นของ JSON

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

2.3 ด้าน User Interface

ในส่วนนี้จะอธิบายถึงการออกแบบ User Interface ของเว็บแอปพลิเคชัน

2.3.1 Design Thinking

กระบวนการออกแบบ design thinking นั้นมีหลากหลายรูปแบบ ทั้งรูปแบบ 3 ขั้น ไปจนถึง 7 ขั้น 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 กลุ่ม เป้าหมาย หรือเข้าใจสิ่งที่ต้องการแก้ไขเพื่อหาแนวทางที่เหมาะสม และดีที่สุดให้ได้ โดยเริ่มต้นจาก การ เข้า ใจความ สร้างสมมติฐาน กระตุนให้เกิดการใช้ความคิดที่นำไปสู่ความคิด สร้างสรรค์ และวิเคราะห์ ปัญหา ให้ถ่องแท้ เพื่อหาแนวทางที่ชัดเจน นำไปสู่การแก้ไขปัญหาที่ตรงประเด็น และสร้างผลลัพธ์ที่ดีที่สุด
2. Define หรือ กำหนดปัญหาให้ชัดเจน คือ การเข้าใจความต้องการ ปัญหา และวิเคราะห์ข้อมูลเชิงลึก เพื่อคัดกรองหาปัญหาที่แท้จริง กำหนดหรือบ่งชี้ปัญหาอย่างชัดเจน เพื่อที่จะเป็นแนวทางในการปฏิบัติ และ มีทิศทางในการแก้ไขปัญหาอย่างชัดเจน
3. Ideate หรือ ระดมความคิด คือ การนำเสนอแนวคิดต่างๆร่วมกัน ถึงวิธีการแก้ไขปัญหา อย่างไม่มี จำกัด การระดมความคิดความมุ่งมองหลากหลาย และมีหลากหลายแนวทางให้ได้มากที่สุด เพื่อ ให้มีฐานข้อมูลในการนำไปวิเคราะห์และสรุปผล เพื่อนำไปแก้ไขปัญหา โดยไม่จำเป็นต้องเป็นแนวทาง ใดแนวทางหนึ่ง และการระดมความคิดยังช่วยมองให้เห็นปัญหาที่หลากหลายได้มากขึ้น

4. Prototype หรือ สร้างต้นแบบที่เลือก คือ การออกแบบผลิตภัณฑ์หรือนวัตกรรม เพื่อสร้างต้นแบบสำหรับการทดสอบ และนำไปใช้จริง ซึ่งคือ การลงมือปฏิบัติหรือทดลองตามแนวทางการแก้ไขปัญหาที่ได้ กำหนดไว้

5. Test หรือ ทดสอบการแก้ไขปัญหา นำ Prototype ที่เราทำการทำขึ้นมาไปทดสอบกับผู้ใช้ว่าสามารถแก้ไขปัญหาของ ผู้ใช้ได้หรือไม่ และหลังจากนั้นถ้าหากการแก้ปัญหายังไม่สามารถช่วยแก้ไขได้ หรือ แก้ไขได้ยังไม่ดีพอ ผู้จัดทำจะต้องกลับไปทำการแก้ไขอีกครั้งจนกว่าจะสามารถออกแบบโปรแกรมที่แก้ไขปัญหา ของผู้ใช้ได้

บทที่ 3

โครงสร้างและขั้นตอนการทำงาน

ในบทนี้จะกล่าวถึงหลักการ และการออกแบบระบบ

3.1 หลักการทำงานของระบบ

3.1.1 ภาพรวมของระบบ (System Overview)

ภาพรวมการทำงานของระบบนี้ จะมีส่วนการทำงานหลัก ๆ ดังนี้

- ระบบการอัปโหลดและการนำযารอยโรคในช่องปากโดย AI
- ระบบประวัติการอัปโหลดและการนำยารอยโรคในช่องปากสำหรับแต่ละผู้ใช้งาน
- ระบบการวินิจฉัยโรคสำหรับทันตแพทย์(Annotation)
- ระบบการแพทย์ทางไกล (Telemedicine) สำหรับให้ความคิดเห็นระหว่างทันตแพทย์ผู้เชี่ยวชาญกับทันตแพทย์ทั่วไป, อาสาสมัครสาธารณสุขประจำหมู่บ้าน, ทันตบุคลากรและผู้ใช้ทั่วไป
- ระบบจัดการผู้ใช้งาน (User Management) และระบบติดตามผู้ใช้งาน (User Tracking)
- ระบบจัดการข้อมูล (Data Management)

3.1.2 โครงสร้างฐานข้อมูล (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 โดยสามารถบอกรายละเอียดได้ว่าเป็นด้วยหรือไม่เป็นด้วย ด้วยเหตุผลอย่างไร

ตาราง users log เป็น table ที่สามารถดูได้เฉพาะ user ที่เป็นผู้ดูแลระบบเท่านั้น โดยจะบอกรายละเอียด action ของผู้ใช้งานทุกคนว่า ได้มี interact กับระบบอย่างไรบ้าง โดยที่ action จะมี login, logout , delete รวมถึง action ที่สำคัญต่าง ๆ ไม่ว่าจะเป็น delete user, การ upload image โดยเก็บเป็น unixtime เพื่อทำให้ง่ายต่อการ parse

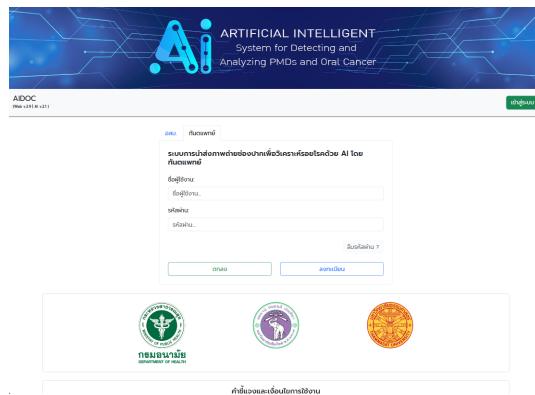
การเข้ารหัสข้อมูล (Encryption) เนื่องจากระบบของเรา มีข้อมูลส่วนตัวมากมายที่เก็บเอาไว้ เราจึงต้องทำการเข้ารหัสข้อมูลตาม requirement ที่เราได้รับมา โดยจะมี file รูปภาพที่เป็นความลับของคนใช้และทำการเข้ารหัส password เพื่อความปลอดภัย

3.2 ส่วนเชื่อมต่อระหว่างผู้ใช้งานกับระบบ (User Interface)

ขอแสดงเฉพาะส่วนที่สำคัญและส่วนที่ผู้จัดทำได้มีส่วนในการพัฒนา

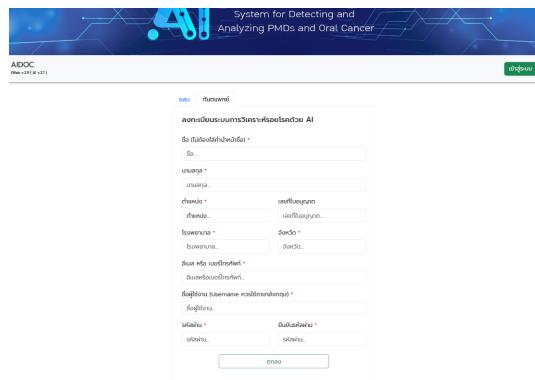
3.2.1 ทันตบุคลากรและผู้ใช้ทั่วไป (User)

หน้าเข้าสู่ระบบ (Login Page)



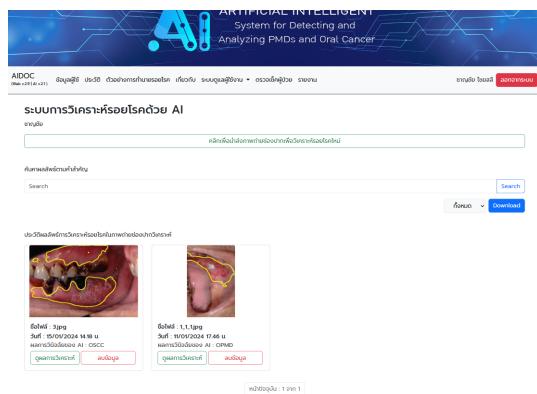
รูปที่ 3.1: login page

หน้าลงทะเบียน (Register Page)

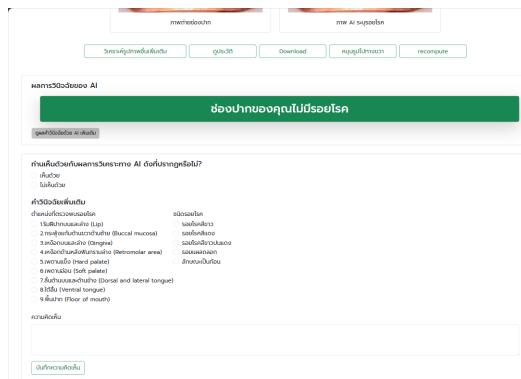


รูปที่ 3.2: register page

หน้าประวัติการอัพโหลดรูปภาพ (History Page)



รูปที่ 3.3: history page



รูปที่ 3.4: history page: comment

หน้าอัปโหลดรูปภาพ (Upload Page)



รูปที่ 3.5: upload page

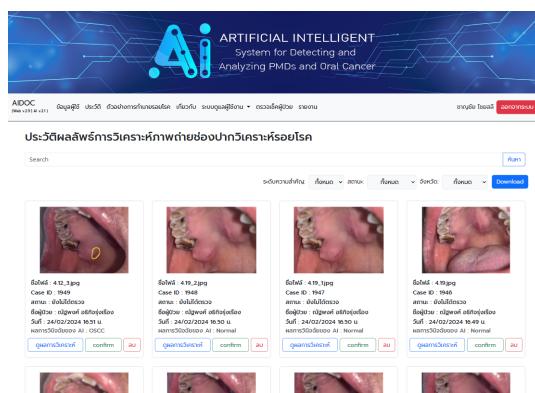


รูปที่ 3.6: preview page



รูปที่ 3.7: uploaded page

หน้าประวัติการส่งรูปจากผู้ป่วย (Checkup Page)

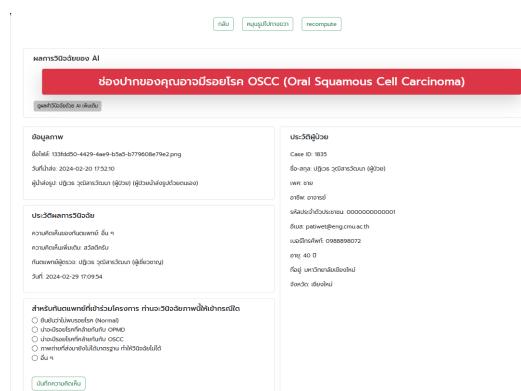


รูปที่ 3.8: checkup page

หน้าตรวจเช็คผู้ป่วย (Checking Page)



รูปที่ 3.9: checking page



รูปที่ 3.10: checking page: comment

หน้ารายงานผลการส่งภาพ (Report Page)

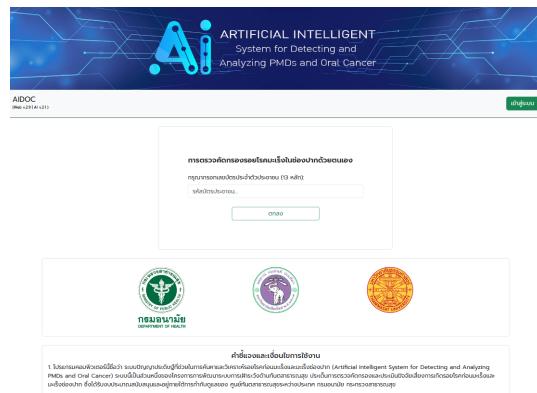
รายงานผลการส่งร่างกายวินิจฉัย										
กลุ่มตัวอย่าง	จำนวน	ผลการตรวจโดย AI			ผลการตรวจโดยแพทย์			ความถูกต้องของผลการตรวจ		
		OPIMD	OSSCC	NORMAL	OPIMD	OSSCC	NORMAL	แพทย์ที่ตรวจ	ผู้ตรวจ	จำนวน
บุตรหลาน	1612	266	57	1289	35	15	102	26	15	99.49%
เด็ก	3	0	0	3	0	1	0	2	0	99.94%
รวม	1615	266	57	1292	35	16	102	28	15	99.72%

กลุ่มตัวอย่าง	จำนวน	ผลการตรวจโดย AI			ผลการตรวจโดยแพทย์			ความถูกต้องของผลการตรวจ		
		OPIMD	OSSCC	NORMAL	OPIMD	OSSCC	NORMAL	แพทย์ที่ตรวจ	ผู้ตรวจ	จำนวน
บุตรหลาน	136	53	12	71	6	0	124			
บุตรหลานที่ไม่พบ/Others	72	52	9	11	2	1	69			
เด็ก	31	7	1	23	2	0	29			
เด็กที่ไม่พบ/Others	66	13	3	50	8	1	57			
รวมทั้งหมด	8	4	9	5	0	0	8			

รูปที่ 3.11: report page

3.2.2 ประชาชนทั่วไป (General public)

หน้าเข้าสู่ระบบ (Login Page)



รูปที่ 3.12: login page

หน้าลงทะเบียน (Register Page)

รูปที่ 3.13: register page

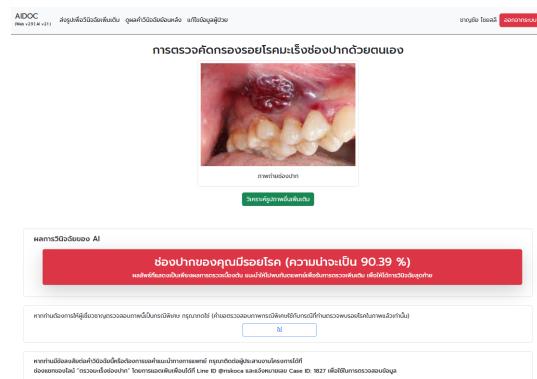
หน้าอัพโหลดรูปภาพ (Upload Page)



รูปที่ 3.14: upload page



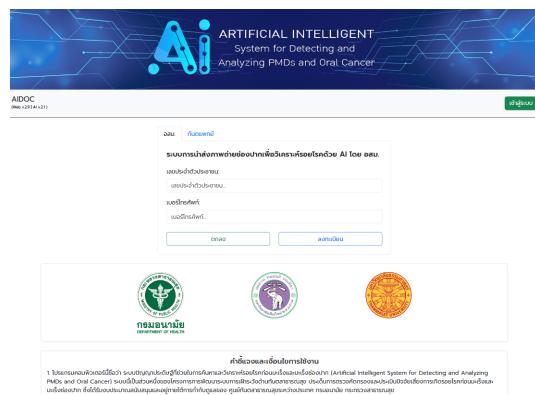
รูปที่ 3.15: preview page



รูปที่ 3.16: uploaded page

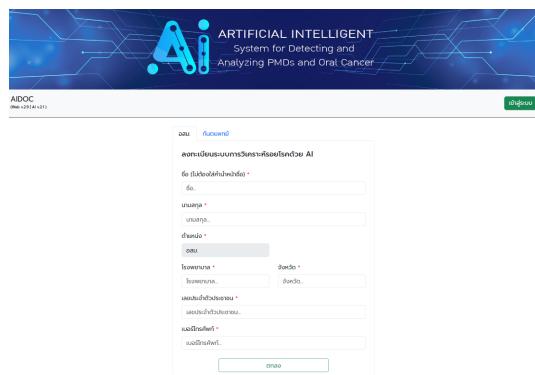
3.2.3 อาสาสมัครสาธารณสุขประจำหมู่บ้าน (อสม.)

หน้าเข้าสู่ระบบ (Login Page)



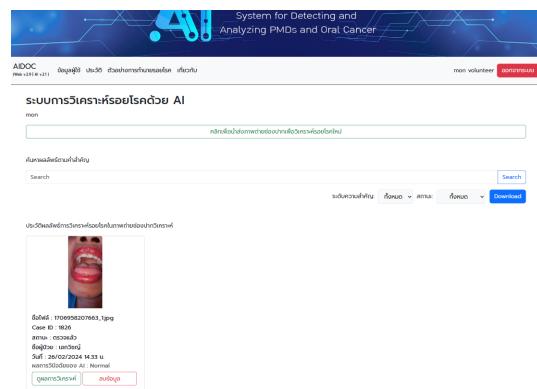
รูปที่ 3.17: login page

หน้าลงทะเบียน (Register Page)



รูปที่ 3.18: register page

หน้าประวัติการอัปโหลดรูปภาพ (History Page)



รูปที่ 3.19: history page

หน้าอัปโหลดรูปภาพ (Upload Page)



รูปที่ 3.20: upload page



รูปที่ 3.21: preview page



ຮູບທີ 3.22: uploaded page

บทที่ 4

การทดลองและผลลัพธ์

ในบทนี้จะทดสอบเกี่ยวกับการทำงานในฟังก์ชันหลักๆ

4.1 การทดลองเกี่ยวกับการทำงานของระบบ

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

4.1.1 ผู้ใช้ทั่วไป

ผู้ใช้ทั่วไปมีความต้องการใช้งานระบบที่เรียบง่าย ใช้งานง่าย ไม่ซับซ้อน ดังนั้น ในการทดสอบกับผู้ใช้ทั่วไป ควรเน้นการประเมินปัจจัยต่างๆ เช่น

- ความนำใช้งาน: Ease of use
- ความพึงพอใจของผู้ใช้งาน: User satisfaction
- ประโยชน์: Benefits
 - ตัวอย่างวิธีการทดสอบกับผู้ใช้ทั่วไป ได้แก่
 - ให้ผู้ใช้ทดสอบระบบและรวบรวมข้อมูลเกี่ยวกับประสบการณ์การใช้งาน เช่น ระยะเวลาในการดำเนินการแต่ละขั้นตอน ความสะดวกในการใช้งาน เป็นต้น
 - ให้ผู้ใช้ตอบแบบสอบถามเกี่ยวกับความพึงพอใจต่อระบบ เช่น ความง่ายในการใช้งาน ความน่าสนใจ ของเนื้อหา เป็นต้น
 - ให้ผู้ใช้ประเมินประโยชน์ที่ได้รับจากการใช้งาน เช่น ช่วยให้ประหยัดเวลา ช่วยให้เข้าใจข้อมูลต่างๆ ได้ง่าย เป็นต้น

4.1.2 ทันตแพทย์

ทันตแพทย์มีความต้องการใช้งานระบบที่มีประสิทธิภาพ ถูกต้องแม่นยำ และสามารถซ่วยในตรวจคัดกรองมะเร็งช่องปากได้อย่างมีประสิทธิภาพ ดังนั้น ในการทดสอบกับทันตแพทย์ ควรเน้นการประเมินปัจจัยต่างๆ เช่น

- ความนำใช้งาน: Ease of use
- ความพึงพอใจของผู้ใช้งาน: User satisfaction
- ประโยชน์: Benefits
 - การช่วยในการตรวจคัดกรองมะเร็งช่องปาก: Screening
 - ตัวอย่างวิธีการทดสอบกับทันตแพทย์ ได้แก่
 - ให้ทันตแพทย์ทดสอบระบบภายใต้สถานการณ์จริง เช่น ถ่ายภาพช่องปากของผู้ป่วย และให้ระบบตรวจคัดกรอง และให้ทันตแพทย์ประเมินความถูกต้องแม่นยำของระบบ เป็นต้น
 - ให้ทันตแพทย์ประเมินประโยชน์ที่ได้รับจากการใช้งาน เช่น ช่วยให้ตรวจคัดกรองมะเร็งช่องปากได้อย่างมีประสิทธิภาพหรือไม่ เป็นต้น

4.1.3 ทันตบุคลากร

ทันตบุคลากรมีความต้องการใช้งานระบบที่อำนวยความสะดวกในการทำงาน เช่น การดูประวัติการตรวจคัดกรองมะเร็งช่องปาก การบันทึกข้อมูล การสรุปผลการตรวจคัดกรองมะเร็งช่องปาก ดังนั้น ในการทดสอบกับทันตบุคลากร ควรเน้นการประเมินปัจจัยต่างๆ เช่น

- ความสะดวกในการใช้งาน: Ease of use
- ประโยชน์: Benefits

ตัวอย่างวิธีการทดสอบกับทันตบุคลากร ได้แก่

- ให้ทันตบุคลากรทดสอบระบบและรับรวมข้อมูลเกี่ยวกับประสบการณ์การใช้งาน เช่น ระยะเวลาในการดำเนินการแต่ละขั้นตอน ความสะดวกในการใช้งาน เป็นต้น
- ให้ทันตบุคลากรประเมินประโยชน์ที่ได้รับจากระบบ เช่น ระบบช่วยให้ทำงานได้อย่างมีประสิทธิภาพ หรือไม่ เป็นต้น

4.1.4 อาสาสมัครสาธารณสุขประจำหมู่บ้าน (อสม.)

อสม. มีความต้องการใช้งานระบบที่เข้าใจง่าย ใช้งานสะดวก และสามารถช่วยให้บริการประชาชนได้อย่างมีประสิทธิภาพ ดังนั้น ในการทดสอบกับอสม. ควรเน้นการประเมินปัจจัยต่างๆ เช่น

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

ตัวอย่างวิธีการทดสอบกับอสม. ได้แก่

- ให้อสม.ทดสอบระบบและรับรวมข้อมูลเกี่ยวกับประสบการณ์การใช้งาน เช่น ระยะเวลาในการดำเนินการแต่ละขั้นตอน ความสะดวกในการใช้งาน เป็นต้น
- ให้อสม.ตอบแบบสอบถามเกี่ยวกับความพึงพอใจต่อระบบ เช่น ความง่ายในการใช้งาน ความน่าสนใจของเนื้อหา เป็นต้น
- ให้อสม.ประเมินประโยชน์ที่ได้รับจากระบบ เช่น ระบบช่วยให้บริการประชาชนได้อย่างมีประสิทธิภาพ หรือไม่ เป็นต้น

ทั้งนี้ ในการทดสอบระบบกับผู้ใช้ทั่วไป, ทันตแพทย์, ทันตบุคลากรและอสม. จะพิจารณาจากปัจจัยต่างๆ เช่น วัตถุประสงค์ของการประเมิน ขอบเขตของการประเมิน ความพร้อมของระบบ เป็นต้น เพื่อให้ได้ผลการประเมินที่มีประสิทธิภาพ

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผล

ในการทำโครงการนี้ สามารถพัฒนาเว็บแอปพลิเคชัน ที่สามารถทำงานร่วมกับระบบ AI ได้จริง โดยมีความแม่นยำในการนำนายอร่อยโกรในระดับที่พึงพอใจ แต่สามารถพัฒนาให้ดียิ่งขึ้นกว่านี้ได้อีกทั้งในส่วนของ AI และส่วนของระบบเว็บแอปพลิเคชัน ที่สามารถทำให้มีความพึงพอใจในการใช้งานมากขึ้นได้ เพิ่มให้มีความสวยงามและใช้งานง่ายขึ้น

ในส่วนของการรองรับผู้ใช้งาน ยังไม่สามารถทำได้ดีพอที่จะรองรับผู้ใช้งานหลายคนทั่วทั้งประเทศพร้อมกันได้ เนื่องจากข้อจำกัดของเซิร์ฟเวอร์ที่เชื่อมโยงในปัจจุบัน ไม่ได้มีประสิทธิภาพมากพอ และทางคณะผู้จัดทำได้วางแผนแนวทางการพัฒนาระบบ ต่อไปในอนาคตเพื่อให้มีความเสถียรและสามารถขยายขนาด (scalability) ได้โดยไม่ใช้ทรัพยากรามากเกินจำเป็น

5.2 ปัญหาที่พบและแนวทางการแก้ไข

ในการทำโครงการนี้ พบร่วมกับปัญหาหลักๆ ดังนี้

- ระบบเดิมไม่ได้มีการออกแบบเพื่อรองรับการเพิ่มฟีเจอร์ใหม่ตั้งแต่แรก ทำให้การพัฒนามีความล่าช้ากว่าที่ควร เพราะต้องปรับโครงสร้างระบบใหม่ เพื่อให้การพัฒนาระบบในอนาคตเป็นไปได้อย่างราบรื่นขึ้น
- คณะผู้จัดทำไม่ได้มีพื้นฐานความรู้ด้านการแพทย์ จึงทำให้ไม่ค่อยเข้าใจมุมมองในฐานะทันตแพทย์ ว่า ต้องการให้ระบบทำอะไรได้บ้าง
- ในช่วงที่ทำการพัฒนาระบบอยู่ มีการใช้งานจากผู้ใช้จริงอยู่ด้วย จึงมีความเสี่ยงที่เมื่อเกิดความผิดพลาด จะทำให้ข้อมูลจงใจเสียหาย
- ในปัจจุบันพัฒนามาเพื่อรองรับเฉพาะ web browser อาจจะทำให้การแสดงผลไม่ถูกต้องในบางอุปกรณ์ ที่มีขนาดหน้าจอไม่ตรงกับที่ระบบรองรับ

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

ข้อเสนอแนะเพื่อพัฒนาโครงการนี้ต่อไป มีดังนี้

- ปรับโครงสร้างระบบโดยย้ายระบบไปยัง cloud จะทำให้มี scalability และ reliability ที่สูงขึ้น
- มีระบบ backup และ rollback ฐานข้อมูล เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นได้ระหว่างการพัฒนาระบบ
- ควรมีการทำระบบอีกเวอร์ชันเพิ่มอีกมาเพื่อให้เชื่อมอุปกรณ์เคลื่อนที่โดยเฉพาะ เพื่อให้ใช้งานง่ายและสวยงามขึ้น
- ควรเพิ่มระบบ security ให้ดียิ่งกว่าปัจจุบัน เพราะมีข้อมูลส่วนตัวของผู้ใช้เก็บอยู่ด้วย อาจเกิดความเสียหายขึ้นได้เมื่อมีผู้ไม่หวังดีมาโจมตีระบบ
- ควรมีการทำ document ไว้เพื่อให้ผู้ที่เข้ามาพัฒนาระบบต่อไปสามารถเข้าใจได้ง่ายขึ้น