



การจัดการความรู้ (Knowledge Management)
การบริการข้อมูลด้านอุตุนิยมวิทยาการบินด้วยรูปแบบ
ดิจิทัลพื้นที่ภาคใต้ฝั่งตะวันออก
ศูนย์อุตุนิยมวิทยาภาคใต้ฝั่งตะวันออก
ปีงบประมาณ 2568



รายงานสภาพอากาศ
บริเวณสนามบินหาดใหญ่ (VTSS)
วันพฤหัสบดีที่ 14 สิงหาคม 2568 เวลา 15:30 น.

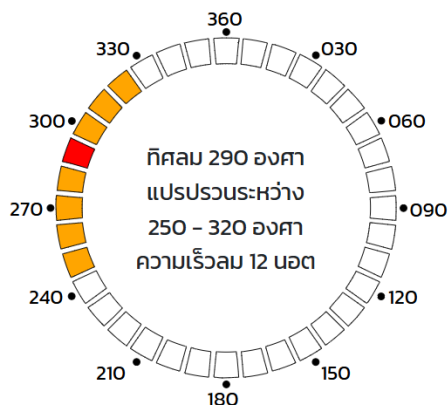
ข่าวอากาศการบินล่าสุด (METAR/SPECI)

METAR VTSS 140830Z 29012KT 250V320 9999 SCT020 SCT100 31/22 Q1007 NOSIG=

ข่าวพยากรณ์อากาศ (TAF)

TAF VTSS 140500Z 1406/1506 25008KT 9999 SCT020 BKN100
TEMPO 1406/1412 25015G25KT -TSRA FEW016CB SCT018 BKN100
BECMG 1415/1417 19003KT
BECMG 1503/1505 25008KT=

ทิศทางและความเร็วลม



สภาพอากาศ



มีเมฆบางส่วน
ทัศนวิสัย 10 กิโลเมตร

คำนำ

ด้วยศูนย์อุตุนิยมวิทยาภาคใต้ฝั่งตะวันออก (ศอ.) ได้จัดทำองค์ความรู้ (KM) เรื่อง “การบริการข้อมูลด้านอุตุนิยมวิทยาการบินด้วยรูปแบบดิจิทัลพื้นที่ภาคใต้ฝั่งตะวันออก” เพื่อแปลรหัสข่าวอากาศการบิน METAR SPECI และ TAF ให้อยู่ในรูปแบบที่เข้าใจได้ง่าย ทำให้ผู้รับบริการ และผู้ที่สนใจ สามารถเข้าถึงและเข้าใจข่าวอากาศการบินได้รวดเร็วยิ่งขึ้น

อนึ่ง ในการจัดทำครั้งนี้ คณะผู้จัดทำ ได้รวบรวมแนวทางในการออกแบบและจัดทำเว็บไซต์ รวมถึงการเผยแพร่เว็บไซต์สู่สาธารณะ โดยประยุกต์ใช้ปัญญาประดิษฐ์ช่วยในการออกแบบและสร้างชุดคำสั่ง และแนะนำการใช้งานเว็บไซต์ที่เป็นผลผลิตในการจัดการความรู้ในครั้งนี้

ทั้งนี้หวังเป็นอย่างยิ่งว่าองค์ความรู้ฉบับนี้จะเป็นประโยชน์กับทั้งศูนย์อุตุนิยมวิทยาภาคใต้ฝั่งตะวันออก หน่วยงานที่เกี่ยวข้อง และผู้ที่สนใจ

คณะผู้จัดทำ

ศูนย์อุตุนิยมวิทยาภาคใต้ฝั่งตะวันออก

สิงหาคม 2568

สารบัญ

	หน้า
คำนำ	ก
สารบัญ	๗
สารบัญตาราง	ค
สารบัญภาพ	ง
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขต	2
1.4 วิธีการดำเนินงาน	3
1.5 ผลที่คาดว่าจะได้รับ	3
บทที่ 2 การตรวจและรายงานอากาศการบิน	4
บทที่ 3 การใช้ปัญญาประดิษฐ์ช่วยในการทำเว็บไซต์ และการเผยแพร่เว็บไซต์ทางอินเทอร์เน็ต	11
3.1 ปัญญาประดิษฐ์	11
3.2 การใช้ปัญญาประดิษฐ์ช่วยในการทำเว็บไซต์	14
3.3 ตัวอย่างการใช้ ChatGPT ในการทำเว็บไซต์	14
3.4 แนะนำเครื่องมือเพิ่มเติมสำหรับทำเว็บไซต์	18
3.5 การพัฒนาเว็บไซต์	20
3.6 การนำเว็บไซต์ไปเผยแพร่ (Website Deployment) ผ่านผู้ให้บริการ Hosting	26
บทที่ 4 การพัฒนาเว็บไซต์สำหรับรายงานสภาพอากาศบริเวณสนามบิน	30
4.1 การพัฒนาเว็บไซต์รายงานสภาพอากาศบริเวณสนามบิน	30
4.2 ข้อมูลนำเข้า	31
4.3 การประมวลผลข้อมูลและการสร้างหน้าเว็บไซต์	34
บทที่ 5 การใช้งานเว็บไซต์	38
5.1 สนามบินในพื้นที่ภาคใต้ฝั่งตะวันออก	38
5.2 สนามบินอื่นๆ	46
5.3 ข้อมูลเพิ่มเติม	50
5.4 ภูมิอากาศสนามบิน (Aerodrome Climatological)	50
5.5 การใช้งานบนโทรศัพท์มือถือ	51
บรรณานุกรม	53
ภาคผนวก	54
ภาคผนวก ก. แนะนำผู้รับบริการสำหรับจัดเก็บข้อมูลบนคลาวด์ (Cloud Storage)	55
ภาคผนวก ข. การสร้างตัวกลาง (Proxy) สำหรับแก้ปัญหา CORS ในเว็บไซต์	63

สารบัญตาราง

	หน้า
ตารางที่ 2.1 อักษรย่อที่ใช้ใน Present Weather Abbreviations ในการรายงาน METAR	7
ตารางที่ 4.1 กราฟิกสภาพอากาศปัจจุบัน (Present Weather)	37

สารบัญภาพ

	หน้า
ภาพที่ 2.1 แสดงรหัส METAR/SPECI ในรูปแบบมาตรฐานตามเอกสาร WMO No. 306	5
ภาพที่ 3.1 แสดงตัวอย่าง Generative AI ในด้านต่างๆ	13
ภาพที่ 3.2 หน้าเว็บไซต์ https://chatgpt.com	14
ภาพที่ 3.3 แสดงโค้ดภาษา HTML พร้อมกับคำอธิบายที่ ChatGPT สร้างขึ้นมา	16
ภาพที่ 3.4 แสดงผลลัพธ์ของโค้ดภาษา HTML ที่ ChatGPT สร้างขึ้นมา	16
ภาพที่ 3.5 แสดงผลลัพธ์เพิ่มเติมที่ ChatGPT สร้างขึ้นมา	17
ภาพที่ 3.6 แสดงโปรแกรม Visual Studio Code	18
ภาพที่ 3.7 แสดงเว็บไซต์ https://getbootstrap.com	19
ภาพที่ 3.8 แสดงเว็บไซต์ https://api.jquery.com	19
ภาพที่ 3.9 แสดงระบบหน้าบ้าน (Frontend) และระบบหลังบ้าน (Backend)	20
ภาพที่ 3.10 แสดงโครงสร้างของ HTML DOM	24
ภาพที่ 3.11 แสดงไฟล์ HTML, JavaScript และ CSS ที่ต้องการเผยแพร่	27
ภาพที่ 3.12 แสดงเว็บไซต์ https://dash.cloudflare.com	28
ภาพที่ 3.13 แสดงขั้นตอนการตั้งชื่อเว็บไซต์	28
ภาพที่ 3.14 แสดงขั้นตอนการอัปโหลดและเผยแพร่เว็บไซต์	29
ภาพที่ 3.15 แสดงเว็บไซต์ที่เผยแพร่	29
ภาพที่ 4.1 แสดงแผนผังข้อมูลนำเข้า การประมวลผล และผลลัพธ์ที่ต้องการ ในการพัฒนาเว็บเพื่อ รายงานสภาพอากาศบริเวณสนามบิน	30
ภาพที่ 4.2 แสดงเว็บไซต์ https://aviationweather.gov	31
ภาพที่ 4.3 แสดง API ที่ aviationweather.gov ให้บริการ	31
ภาพที่ 4.4 แสดงการเรียก API บนเว็บไซต์ aviationweather.gov	32
ภาพที่ 4.5 แสดงเว็บไซต์ https://nsweb.tmd.go.th/#showMetars	33
ภาพที่ 4.6 แสดงเว็บไซต์ https://weather.tmd.go.th	33
ภาพที่ 4.7 แสดงโปรแกรม Visual Studio Code	36
ภาพที่ 4.8 แผนภาพแสดงเครื่องมือ และขั้นตอนการดำเนินการทำเว็บไซต์	36
ภาพที่ 5.1 แสดงเว็บไซต์ https://aero.semet.uk	38
ภาพที่ 5.2 แสดงข่าวรายงานสภาพอากาศ (METAR/SPECI) ที่ปรากฏขึ้นมา	39
ภาพที่ 5.3 แสดงสีพื้นหลังของการรายงานข่าว METAR (บน) และ SPECI (ล่าง)	40
ภาพที่ 5.4 แสดงเว็บไซต์ส่วนแสดงการแปลข่าวอากาศการบิน (METAR/SPECI)	40
ภาพที่ 5.5 แสดงปุ่มกด เพื่อแสดงการแปลข่าวพยากรณ์อากาศบริเวณสนามบิน (TAF)	41
ภาพที่ 5.6 แสดงการแปลข่าวพยากรณ์อากาศบริเวณสนามบิน (TAF)	41

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 5.7 แสดงการเน้น (เปลี่ยนสี) แบบจับคู่ของเส้นเวลา และการแปลข่าว TAF	42
ภาพที่ 5.8 แสดงการคาดการณ์การเปลี่ยนแปลงแบบ BECMG	42
ภาพที่ 5.9 แสดงเว็บไซต์ส่วนภาพเรดาร์ตรวจอากาศ บริเวณสนามบินนราธิวาส	43
ภาพที่ 5.10 แสดงภาพเรดาร์ตรวจอากาศแบบเต็มจอ บริเวณสนามบินหาดใหญ่	44
ภาพที่ 5.11 แสดงเว็บไซต์ส่วนแสดงข่าว METAR ย้อนหลัง 24 ชั่วโมง	45
ภาพที่ 5.12 แสดงหน้าเว็บไซต์ข่าวอากาศการบินสำหรับสนามบินอื่นๆ	46
ภาพที่ 5.13 แสดงปุ่มตั้งค่าเรดาร์	46
ภาพที่ 5.14 แสดงหน้าเว็บไซต์เรดาร์ตรวจอากาศสำหรับสนามบินอื่นๆ	47
ภาพที่ 5.15 แสดงแบบฟอร์มการตั้งค่าภาพเรดาร์	47
ภาพที่ 5.16 แสดงหน้าเว็บไซต์ตัวช่วยตั้งค่าภาพเรดาร์	48
ภาพที่ 5.17 แสดงข้อมูลเพิ่มเติมสำหรับอุตุนิยมวิทยาการบิน	50
ภาพที่ 5.18 แสดงขั้นตอนไปยังหน้าเว็บไซต์ภูมิอากาศสนามบิน	50
ภาพที่ 5.19 แสดงหน้าเว็บไซต์ภูมิอากาศสนามบิน	51
ภาพที่ 5.20 แสดงการติดตั้งเว็บไซต์เป็นแอปพลิเคชันแบบ PWA	52
ภาพที่ 5.21 แสดงการใช้งานเว็บไซต์บนโทรศัพท์มือถือ	52

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ข่าวอากาศการบินเป็นข่าวที่มีความสำคัญและจำเป็นในกิจการการบิน การทราบถึงสภาพอากาศจะช่วยเสริมสร้างความปลอดภัยในการบิน การตัดสินใจที่มีประสิทธิภาพในการวางแผนการบินและการจัดการสนามบิน รวมถึงการพยากรณ์อากาศและการเตรียมความพร้อมสำหรับเหตุการณ์ทางอากาศที่อาจเกิดขึ้น ทำให้มีประสิทธิภาพในการจัดการการบิน

โดยองค์การการบินพลเรือนระหว่างประเทศ (International Civil Aviation Organization: ICAO) และองค์การอุตุนิยมวิทยาโลก (World Meteorological Organization: WMO) ได้กำหนดและควบคุมมาตรฐานในการให้บริการข้อมูลสภาพอากาศสำหรับการบินระหว่างประเทศ ไว้ในคู่มือ “Meteorological Service for International Air Navigation (Annex 3)” เป็นคู่มือในด้านการจัดทำรายงานสภาพอากาศ การพยากรณ์อากาศและการบริการข้อมูลที่เกี่ยวข้อง รวมถึงมาตรการในการฝึกอบรมและพัฒนาบุคลากรในด้านการบริการสภาพอากาศ จึงช่วยให้ประเทศต่าง ๆ สามารถแลกเปลี่ยนข้อมูลและทำงานร่วมกันได้อย่างมีประสิทธิภาพ

ข่าว METAR และ SPECI เป็นข่าวรายงานสภาพอากาศที่ใช้ในการบินและการเฝ้าระวังสภาพอากาศ ที่ถูกระบุไว้ใน Annex 3 โดยข่าว METAR คือ ข่าวรายงานสภาพอากาศที่ออกเป็นประจำทุกครึ่งชั่วโมงหรือหนึ่งชั่วโมง ใช้สำหรับให้ข้อมูลเกี่ยวกับสภาพอากาศบริเวณสนามบิน รวมถึงการพยากรณ์แนวโน้มสภาพอากาศ มีวัตถุประสงค์เพื่อนำไปใช้สำหรับประกอบการวางแผนการบิน (Flight Plan) และใช้ในการออกประกาศระบบ VOLMET และ D-VOLMET สำหรับการเผยแพร่กระจายข่าวออกไปภายนอกท่าอากาศยาน และข่าว SPECI คือข่าวรายงานสภาพอากาศที่จะออกเมื่อมีการเปลี่ยนแปลงสภาพอากาศที่สำคัญ เช่น การเปลี่ยนแปลงของลม ทิศวิสัย หรือปรากฏการณ์ทางอากาศอื่น ๆ ที่เกิดขึ้นระหว่างช่วงเวลารายงาน METAR ช่วยให้ผู้ใช้ข้อมูลสามารถรับรู้ถึงสภาพอากาศที่เปลี่ยนแปลงได้อย่างทันทั่วถึง

ข่าวรายงานอากาศการบิน (METAR / SPECI) นั้น ถูกรายงานเป็นรหัสมาตรฐาน ตามรูปแบบที่ระบุไว้ในเอกสารขององค์การอุตุนิยมวิทยาโลก “Manual on Codes International Codes Volume I.1 (WMO No. 306)” โดยข่าว METAR เต็มรูปแบบมีได้ถึง 18 กลุ่ม ซึ่งหน่วยงานอุตุนิยมวิทยาของแต่ละประเทศ กำหนดให้มีการบังคับใช้รหัสและวิธีปฏิบัติในประเทศนั้นๆ สำหรับประเทศไทยได้กำหนดกฎและวิธีการเข้ารหัส ไว้ใน “คู่มือมาตรฐาน การตรวจและรายงานอากาศการบินแบบ METAR และ SPECI” โดยกองอุตุนิยมวิทยาการบิน กรมอุตุนิยมวิทยา จะแบ่งข่าวออกเป็น 13 ส่วน และสำหรับส่วนที่ 13 “การพยากรณ์แนวโน้มสภาวะอากาศบริเวณสนามบิน (Trend forecast)” กรมอุตุนิยมวิทยาได้กำหนดกฎและวิธีการเข้ารหัส ไว้ใน “คู่มือมาตรฐาน การพยากรณ์แนวโน้มสภาวะอากาศบริเวณสนามบิน”

จะเห็นได้ว่าการเข้ารหัสข่าว ข่าวรายงานอากาศการบิน (METAR / SPECI) และข่าวพยากรณ์อากาศบริเวณสนามบิน (TAF) มีความซับซ้อน และเฉพาะเจาะจง ถ้าไม่ได้เป็นผู้ที่มีพื้นฐานด้านการบิน หรือปฏิบัติงานเกี่ยวกับกิจการการบิน/อากาศการบิน ก็อาจไม่เข้าใจและไม่สามารถแปลความหมายรหัสข่าวได้ และถึงแม้จะมีพื้นฐานเรื่องรหัสข่าวอากาศการบินอยู่บ้าง แต่ด้วยความหลากหลายของรหัสข่าว และสภาพอากาศของแต่ละพื้นที่ต่างกัน ก็อาจจะลืมหสบางส่วนที่ไม่ค่อยถูกใช้งานได้

ศูนย์อุตุนิยมวิทยาภาคใต้ฝั่งตะวันออก จึงเห็นว่าการแปลรหัสข่าวรายงานอากาศการบิน (METAR / SPECI) และข่าวพยากรณ์อากาศบริเวณสนามบิน (TAF) ให้อยู่ในรูปแบบดิจิทัลนั้น มีความสำคัญ นอกจากจะช่วยให้ผู้ที่ไม่มีความรู้เฉพาะด้านสามารถเข้าใจข้อมูลสภาพอากาศได้อย่างรวดเร็วและชัดเจนแล้ว ยังช่วยในการอธิบายข้อมูลที่ซับซ้อน โดยลดการใช้ข้อความและทำให้ข้อมูลน่าสนใจยิ่งขึ้น และเป็นการใช้งานข้อมูลที่คุ้มค่า ที่ไม่ได้ใช้งานเฉพาะในสนามบินเพียงอย่างเดียว ประชาชนที่อยู่บริเวณใกล้เคียงสนามบินหรือผู้ที่ต้องการเดินทางโดยอากาศยาน ยังสามารถใช้ประโยชน์ในการวางแผนสภาพอากาศได้อย่างมีประสิทธิภาพ เนื่องจากเป็นข่าวที่ถูกรายงานโดยเครื่องมือที่มีมาตรฐาน และบุคลากรของกรมอุตุนิยมวิทยา ทำให้ข้อมูลมีความน่าเชื่อถือ

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาองค์ความรู้ที่ใช้ในการทำเว็บไซต์ รวมถึงการเผยแพร่เว็บไซต์สู่สาธารณะจากแหล่งความรู้ทางอินเทอร์เน็ต และประยุกต์ใช้ปัญญาประดิษฐ์ช่วยออกแบบเว็บไซต์

1.2.2 เพื่อเผยแพร่ข้อมูลอากาศการบินในรูปแบบดิจิทัล ให้ผู้ที่สนใจและต้องการวางแผนด้านสภาพอากาศ ให้เข้าใจข้อมูลอากาศการบินได้รวดเร็วและชัดเจนยิ่งขึ้น

1.3 ขอบเขต

1.3.1 ทบทวนความรู้ทางอุตุนิยมวิทยาการบินและมาตรฐานการเข้ารหัสข่าวอากาศการบิน

1.3.2 ออกแบบเว็บไซต์สำหรับเผยแพร่ข้อมูลสภาพอากาศบริเวณสนามบิน โดยใช้ข่าวรายงานอากาศการบิน (METAR / SPECI) และข่าวพยากรณ์อากาศบริเวณสนามบิน (TAF) โดยมีภาพเรดาร์ตรวจอากาศในเขตรัศมี 64 กิโลเมตรรอบสนามบิน สำหรับสนามบินในพื้นที่ภาคใต้ฝั่งตะวันออกจำนวน 8 แห่ง ประกอบไปด้วย สนามบินหัวหิน (VTPH) สนามบินชุมพร (VTSE) สนามบินสมุย (VTSM) สนามบินสุราษฎร์ธานี (VTSB) สนามบินนครศรีธรรมราช (VTSF) สนามบินหาดใหญ่ (VTSS) สนามบินนราธิวาส (VTSC) และ สนามบินเบตง (VTSY) ส่วนสนามบินอื่นๆ ทั่วประเทศ สามารถตั้งค่าภาพเรดาร์ได้ด้วยตนเอง

1.4 วิธีการดำเนินงาน

วิธีการดำเนินการจัดการความรู้มีดังต่อไปนี้

- 1.4.1 การบ่งชี้ความรู้
- 1.4.2 การสร้างและแสวงหาความรู้
- 1.4.3 การจัดความรู้ให้เป็นระบบ
- 1.4.4 การประมวลและกลั่นกรองความรู้
- 1.4.5 การเข้าถึงความรู้
- 1.4.6 การแบ่งปันแลกเปลี่ยนเรียนรู้
- 1.4.7 การเรียนรู้

1.5 ผลที่คาดว่าจะได้รับ

- 1.5.1 มีเว็บไซต์สำหรับเผยแพร่ข้อมูลสภาพอากาศบริเวณสนามบิน ช่วยให้ผู้ที่สนใจและต้องการวางแผนด้านสภาพอากาศ ให้เข้าใจข้อมูลอากาศการบินได้รวดเร็วและชัดเจนยิ่งขึ้น
- 1.5.2 ช่วยเพิ่มประสิทธิภาพการทำงานให้แก่ผู้ที่ปฏิบัติงานด้านอากาศการบิน
- 1.5.3 มีองค์ความรู้ในการจัดทำและเผยแพร่เว็บไซต์ ที่อาจเป็นประโยชน์ในด้านอื่นๆ โดยเฉพาะการเผยแพร่ข้อมูลสาธารณะที่มีการเปลี่ยนแปลงข้อมูลบ่อย ทำให้มีช่องทางการเข้าถึงข้อมูลที่หลากหลาย

บทที่ 2

การตรวจและรายงานอากาศการบิน

รายงานอากาศการบิน METAR (Meteorological Aerodrome report) คือ รูปแบบการรายงานผล การตรวจวัดค่าสารประกอบอุตุนิยมวิทยาและสภาพอากาศที่เกิดขึ้นภายในบริเวณท่าอากาศยาน ซึ่งทำการรายงานเป็นแบบประจำทุกครึ่งชั่วโมงหรือหนึ่งชั่วโมง โดยมีวัตถุประสงค์เพื่อนำไปใช้สำหรับประกอบการวางแผนการบิน (Flight Plan) และใช้ในการออกประกาศระบบ VOLMET และ D-VOLMET สำหรับการ เผยแพร่กระจายข่าวออกไปภายนอกท่าอากาศยาน

เนื้อหาและรูปแบบ (Content and format) ของรายงานอากาศการบิน METAR เป็นไปตามข้อกำหนดขององค์การอุตุนิยมวิทยาโลก (World Meteorological Organization: WMO) และองค์การการบินพลเรือนระหว่างประเทศ (International Civil Aviation Organization: ICAO)

รูปแบบ METAR ประกอบไปด้วยส่วนต่างๆ ดังนี้:

- ส่วนที่ 1. ประเภทของรายงาน (Identification of the type of report)
- ส่วนที่ 2. รหัสท่าอากาศยาน (Location indicator)
- ส่วนที่ 3. เวลาตรวจอากาศการบิน (Time of the Observation)
- ส่วนที่ 4. การตรวจอากาศการบินด้วยระบบอัตโนมัติ (Identification of an automated)
- ส่วนที่ 5. ลมผิวพื้น (Surface wind)
- ส่วนที่ 6. ทิศนวิสัย (Visibility)
- ส่วนที่ 7. พิสัยการมองเห็นบนทางวิ่ง (Runway Visual Range: RVR)
- ส่วนที่ 8. สภาพอากาศปัจจุบัน (Present weather)
- ส่วนที่ 9. เมฆ (Cloud)
- ส่วนที่ 10. อุณหภูมิอากาศและอุณหภูมิจุดน้ำค้าง (Air temperature and Dew-point temperature)
- ส่วนที่ 11. ความกดอากาศ (Atmospheric pressure)
- ส่วนที่ 12. ข่าวสารเพิ่มเติม (Supplementary information)
- ส่วนที่ 13. การพยากรณ์แนวโน้มสภาวะอากาศบริเวณสนามบิน (Trend forecast)

FM 15–XV METAR	Aerodrome routine meteorological report (with or without trend forecast)						
FM 16–XV SPECI	Aerodrome special meteorological report (with or without trend forecast)						
CODE FORM :							
METAR or SPECI	COR	CCCC	YYGGggZ	NIL	AUTO	ddffG _{f_mf_m} $\left\{ \begin{array}{l} \text{KT} \\ \text{or} \\ \text{MPS} \end{array} \right\}$	d _n d _n d _n Vd _x d _x d _x
$\left\{ \begin{array}{l} \text{VVVV} \\ \text{or} \\ \text{CAVOK} \end{array} \right\}$	V _N V _N V _N V _N D _v	RD _R D _R /V _R V _R V _R V _R i		w'w'		$\left\{ \begin{array}{l} \text{NsNsNshshshs} \\ \text{or} \\ \text{VVhshshs} \\ \text{or} \\ \text{NSC} \\ \text{or} \\ \text{NCD} \end{array} \right\}$	
T'T'/T' _d T' _d	QPHPHPHPH	REw'w'	$\left\{ \begin{array}{l} \text{WS RDRDR} \\ \text{or} \\ \text{WS ALL RWY} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{(WTsTs/SS')} \\ \text{or} \\ \text{(WTsTs/HHsHsHs}) \end{array} \right\}$	(RD _R D _R /E _R C _R e _R e _R B _R B _R)		
$\left\{ \begin{array}{l} \text{(TTTTT)} \\ \text{or} \\ \text{NOSIG} \end{array} \right\}$	TTGGgg	ddffG _{f_mf_m}	$\left\{ \begin{array}{l} \text{KT} \\ \text{or} \\ \text{MPS} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{VVVV} \\ \text{or} \\ \text{CAVOK} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{w'w'} \\ \text{or} \\ \text{NSW} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{NsNsNshshshs} \\ \text{or} \\ \text{VVhshshs} \\ \text{or} \\ \text{NSC} \end{array} \right\}$	

ภาพที่ 2.1 แสดงรหัส METAR/SPECI ในรูปแบบมาตรฐานตามเอกสาร WMO No. 306

ที่มา: WMO No. 306 Manual on Codes Volume I.1. 2011 ed. p. A-27.

ส่วนที่ 1. ประเภทของรายงาน ระบุประเภทของรายงานสำหรับรายงานอากาศการบินแบบประจำคือ “METAR” และรายงานอากาศการบินแบบพิเศษคือ “SPECI” และถ้าหากมีการแก้ไขข่าว ให้ระบุ “COR” ต่อท้ายประเภทของรายงาน

ส่วนที่ 2. รหัสท่าอากาศยาน ระบุรหัสท่าอากาศยานด้วยตัวอักษรภาษาอังกฤษ 4 ตัวอักษร ตามรูปแบบ “CCCC” ตามข้อกำหนดของ ICAO ซึ่งกำหนดไว้ในเอกสาร Doc 7910 ตัวอย่างเช่น รหัสท่าอากาศยานนานาชาติหาดใหญ่ คือ VTSS เป็นต้น

ส่วนที่ 3. เวลาตรวจอากาศการบิน ระบุเวลาที่ทำการตรวจอากาศการบิน ตามรูปแบบ “YYGGggZ” ประกอบด้วยวันที่และเวลาเป็นชั่วโมงและนาที ตามเวลามาตรฐานสากล (Coordinate Universal Time: UTC (Z)) ตัวอย่างเช่น ตรวจอากาศการบินวันที่ 28 เวลา 08:30 น. ดังนั้นเวลาตรวจอากาศการบินในรายงาน คือ 280130Z

ส่วนที่ 4. การตรวจอากาศการบินด้วยระบบอัตโนมัติ ถ้าสถานีตรวจอากาศการบินใช้ระบบตรวจอากาศแบบอัตโนมัติจะต้องระบุในรายงานด้วยคำว่า “AUTO”

ส่วนที่ 5. ลมผิวพื้น (Surface wind) รายงานตามรูปแบบ “ddffG_{f_mf_m}KT d_nd_nd_nVd_xd_xd_x” เมื่อ ddd คือ ทิศทางของลมผิวพื้นเฉลี่ย ในช่วง 10 นาทีที่ผ่านมา, ff คือ ความเร็วของลมผิวพื้นเฉลี่ย ในช่วง 10 นาทีที่ผ่านมา, f_mf_m คือ ลมกระโชก (Gust) แรงสูงสุดในช่วง 10 นาทีที่ผ่านมา และ d_nd_nd_nVd_xd_xd_x คือความแปรผันของทิศทางลมผิวพื้น (ตามเข็มนาฬิกา) ในช่วง 10 นาทีที่ผ่านมา

ส่วนที่ 6. ทิศนวิสัย (Visibility) ตามรูปแบบ “VVVV V_NV_NV_NV_ND_V” เมื่อ VVVV คือ ค่า Prevailing Visibility, V_NV_NV_NV_N คือ ค่าทัศนวิสัยต่ำสุด และ D_V คือ ทิศของทัศนวิสัยที่ต่ำสุดตามนियามการมองเห็นทางอุตุนิยมวิทยา (Meteorological Optical Range: MOR) รายงานด้วยตัวอักษรหนึ่งหรือสองตัวโดยระบุด้วยทิศทั้ง 8 ทิศจากจุดอ้างอิงสนามบิน

เมื่อสภาพอากาศเข้าสู่เงื่อนไขที่ต้องรายงาน “CAVOK” จะใช้ CAVOK ในการรายงาน โดยไม่ต้องรายงานกลุ่มทัศนวิสัย โดยเงื่อนไขคือ ทัศนวิสัยต้องมีระยะตั้งแต่ 10 กิโลเมตรขึ้นไป, ไม่มีเมฆที่มีฐานต่ำกว่า 5,000 ฟุต, ไม่มีเมฆ Cumulonimbus (CB) และ/หรือ Towering Cumulus (TCU) และไม่มีสภาพอากาศปัจจุบัน

ส่วนที่ 7. พิสัยการมองเห็นบนทางวิ่ง (Runway Visual Range: RVR) ตามรูปแบบ “RD_RDR/V_RV_RV_RV_R” เมื่อ R คือ ตัวระบุกลุ่ม, DR_R คือ ตัวกำหนดตำแหน่งของทางวิ่งที่อยู่ใกล้จุดที่ตรวจวัดค่า RVR ที่สุด ตามด้วยอักษร L, C หรือ R (ถ้ามีมากกว่าหนึ่งทางวิ่ง) เพื่อแยกแยะระหว่างทางวิ่งซ้าย กลาง ขวา ตามลำดับ และ V_RV_RV_RV_R คือ ค่า RVR ที่รายงานระยะเป็นเมตร

ส่วนที่ 8. สภาพอากาศปัจจุบัน (Present Weather) ตามรูปแบบ “w'w'” เมื่อ w'w' คือตัวแทนของสภาพอากาศปัจจุบัน ซึ่งในการตรวจสภาพอากาศปัจจุบันนั้น จะเกี่ยวข้องกับสภาพอากาศที่เกิดขึ้นในขณะทำการตรวจ บริเวณท่าอากาศยานเท่านั้น ในกรณีที่มีการรายงานปรากฏการณ์ในบริเวณใกล้เคียงกับท่าอากาศยาน (in the vicinity: VC) ภายในระยะประมาณ 8 – 16 กิโลเมตรจากจุดอ้างอิงสนามบิน (Aerodrome Reference Point: ARP) อาจใช้จุดอ้างอิงจากการมองเห็นข้อมูลจากเรดาร์ตรวจสภาพอากาศ รายงานจากนักบิน และ เซ็นเซอร์อัตโนมัติเพื่อช่วยในการตัดสินใจ ถ้าไม่มีสภาพอากาศที่มีนัยสำคัญต่อการปฏิบัติงานด้านการบิน ในขณะทำการตรวจอากาศทั้งใน บริเวณท่าอากาศยานหรือบริเวณใกล้เคียงกับท่าอากาศยาน รหัสกลุ่มนี้จะถูกละเว้นไม่ต้องรายงาน

กลุ่มสภาพอากาศปัจจุบันประกอบด้วยคำย่อตัวอักษรหนึ่งตัว หรือมากกว่าหนึ่งตัว (ดังตารางที่ 2.1) การให้รหัสกลุ่มนี้สามารถระบุถึงความรุนแรง (+, -) ความใกล้เคียงท่าอากาศยาน (VC) บ่งชี้ลักษณะ (Descriptor) และปรากฏการณ์ (Phenomena) ตามลำดับ

ตารางที่ 2.1 อักษรย่อที่ใช้ใน Present Weather Abbreviations ในการรายงาน METAR

คุณสมบัติ				ปรากฏการณ์					
ความรุนแรง หรือบริเวณใกล้เคียง		ลักษณะ		หยาดน้ำฟ้า		สิ่งบดบัง		อื่นๆ	
1		2		3		4		5	
-	เบา	MI	ตื้น	DZ	ฝนละออง	BR	หมอกน้ำค้าง	PO	ลมฝุ่นหมุน
		BC	หยาบ	RA	ฝน	FG	หมอก	SQ	ลมสควอลล์
ไม่มี สัญลักษณ์	ปานกลาง	PR	บางส่วน	SN	หิมะ	FU	ควัน	FC	เมฆวง หรือ พวยน้ำ
		DR	ฟุ้งในระดับ ต่ำ	SG	เม็ดหิมะ	VA	ฝ้าภูเขาไฟ	DS	พายุฝุ่น
+	หนัก/ กำลัง พัฒนา	BL	ปลิว	PL	ลูกปราย น้ำแข็ง	DU	ฝุ่น	SS	พายุทราย
		SH	ฝนชุก	GR	ลูกเห็บ	SA	ทราย		
VC	บริเวณ ใกล้เคียง	TS	พายุฟ้า คะนอง	GS	ชนิดเล็ก และ/หรือ ลูกปราย หิมะ	HZ	ฟ้าหลัว		
		FZ	เยือกแข็ง						

ที่มา: คู่มือมาตรฐาน การตรวจและรายงานอากาศการบินแบบ METAR และ SPECI (2563)

ส่วนที่ 9. เมฆ (Cloud) ตามรูปแบบ “N_sN_sN_shshsh_s(CC)” หรือ “NSC” หรือ “NCD” หรือ “VVhshsh_s” เมื่อ

N_sN_sN_s คือ จำนวนเมฆที่แสดงเป็น FEW (ไม่เกิน 1 - 2 ส่วน), SCT (3 - 4 ส่วน), BKN (5 - 7 ส่วน) และ OVC (เต็มท้องฟ้า 8 ส่วน)

hshsh_s คือ ความสูงของฐานเมฆเหนือท่าอากาศยานรายงานได้ทุก ๆ 100 ฟุต จนถึง 10000 ฟุต

(CC) คือ ชนิดของเมฆที่สามารถใช้รายงานได้เฉพาะ Cumulonimbus (CB) และ/หรือ Towering Cumulus (TCU) ในข่าว METAR

NSC แสดงถึงเมฆที่ไม่มีนัยสำคัญ เช่น ไม่มีเมฆที่มีฐานต่ำกว่า 5000 ฟุต หรือต่ำกว่าระดับ ความสูงต่ำสุดที่อนุญาตให้บิน (ขึ้นอยู่กับว่าอะไรสูงกว่า) และไม่มี Towering Cumulus หรือ Cumulonimbus (TCU หรือ CB)

NCD ไม่ได้รายงานโดยผู้ปฏิบัติงานด้านการตรวจอากาศการบินอากาศ แต่ใช้แทนเมื่อระบบ ตรวจอากาศอัตโนมัติไม่สามารถตรวจพบเมฆ และบ่งบอกว่าระบบไม่สามารถตรวจจับเมฆได้ (no clouds are detected by the system)

VVhshshs คือ ทศนิยมสี่ในแนวตั้ง การรายงาน VV/// หมายความว่าไม่สามารถวัดความสูง การมองเห็นในแนวตั้งได้หรือท้องฟ้าถูกปิดบัง

ส่วนที่ 10. อุณหภูมิอากาศและอุณหภูมิจุดน้ำค้าง (Air temperature and Dew-point temperature) ตามรูปแบบ “T’T/T_dT_d” เมื่อ T’T’ คือ อุณหภูมิอากาศ (องศาเซลเซียส) และ T_dT_d คือ อุณหภูมิจุดน้ำค้าง (องศาเซลเซียส)

ส่วนที่ 11. ความกดอากาศ (Atmospheric pressure) ตามรูปแบบ “QP_HP_HP_HP_H” เมื่อ Q เป็นตัวระบุกลุ่ม และ P_HP_HP_HP_H เป็นความกดอากาศที่ได้รับการหักแก้ที่ระดับน้ำทะเลปานกลาง (QNH)

ส่วนที่ 12. ข่าวสารเพิ่มเติม (Supplementary information)

- สภาพอากาศที่ผ่านมา (Recent weather) ตามรูปแบบ “REW’W” หมายถึงสภาพอากาศที่หยุดไปแล้ว หรือลดความรุนแรงลงจากรายงานครั้งล่าสุด หรือภายในชั่วโมงที่ผ่านมา แล้วแต่ว่าระยะเวลาใดจะสั้นกว่า ใน METAR การรายงานสภาพอากาศที่ผ่านมาจะรายงานหลังจากกลุ่มความกดอากาศ QNH

- การตรวจและรายงานลมเฉือน (Wind Shear) ในกรณีที่เกิดเหตุการณ์ลมเฉือน ในท้องถิ่นควรใช้เท่าที่จำเป็น ข้อมูลเพิ่มเติมเกี่ยวกับการเกิดลมเฉือนจะถูกเพิ่มลงข่าว METAR ต่อท้ายกลุ่มความกดอากาศในรูปแบบ WS RD_RDR หรือ WS ALL RWY

ส่วนที่ 13. การพยากรณ์แนวโน้มสภาวะอากาศบริเวณสนามบิน (Trend forecast) ตามรูปแบบ

$$\left\{ \begin{array}{l} \text{TTTTT} \\ \text{or} \\ \text{NOSIG} \end{array} \right\} \quad \text{TTGGgg} \quad \text{ddfffGf_{fm}m} \quad \left\{ \begin{array}{l} \text{KT} \\ \text{or} \\ \text{MPS} \end{array} \right\} \quad \left\{ \begin{array}{l} \text{VVVV} \\ \text{or} \\ \text{CAVOK} \end{array} \right\} \quad \left\{ \begin{array}{l} \text{w'w'} \\ \text{or} \\ \text{NSW} \end{array} \right\} \quad \left\{ \begin{array}{l} \text{NsNsNshshshs} \\ \text{or} \\ \text{VVhshshs} \\ \text{or} \\ \text{NSC} \end{array} \right\}$$

ถ้าหากคาดว่าจะไม่มีการเปลี่ยนแปลงสารประกอบอุตุนิยมวิทยาที่มีนัยสำคัญ (ตามเกณฑ์) ภายในเวลา 2 ชั่วโมงข้างหน้า จากรายงานข่าว METAR หรือ SPECI ฉบับนั้น ให้ต่อท้ายด้วย “NOSIG”

ถ้าหากคาดว่าจะมีการเปลี่ยนแปลงสารประกอบอุตุนิยมวิทยาที่มีนัยสำคัญ (ตามเกณฑ์) ภายในเวลา 2 ชั่วโมงข้างหน้า จากรายงานข่าว METAR หรือ SPECI ฉบับนั้น ให้ต่อท้ายด้วยรูปแบบรหัสเดียวกัน โดยใช้ กลุ่มคำบอกการเปลี่ยนแปลงนำหน้า ตามด้วยสารประกอบอุตุนิยมวิทยาที่คาดว่าจะมีการเปลี่ยนแปลงที่มี นัยสำคัญ (ตามเกณฑ์) ซึ่งมีการเรียงลำดับที่แน่นอน ดังนี้

1. กลุ่มคำบอกการเปลี่ยนแปลง (Change groups)
2. ลมผิวพื้น (Surface wind)
3. ทิศนวิสัย (Visibility)
4. สภาพอากาศ (Weather)
5. เมฆ (Cloud) หรือ ทิศนวิสัยในแนวตั้ง (Vertical visibility)

1) กลุ่มคำบอกการเปลี่ยนแปลง ตามรูปแบบ “TTTTT TTGGgg” เมื่อ TTTTT คือ คำบอกการเปลี่ยนแปลง ดังนี้

- BECMG (มาจาก Becoming) เป็นการเปลี่ยนแปลงแบบค่อย ๆ เปลี่ยน จะใช้เมื่อคาดว่าจะ สาระประกอบอุตุนิยมวิทยาจะเปลี่ยนแปลงถึงค่าหรือผ่านค่าที่สำคัญที่กำหนดไว้ใน ช่วงเวลาที่กำหนด ด้วย อัตราสม่ำเสมอหรือไม่สม่ำเสมอ โดยการเปลี่ยนแปลงแบบ BECMG นี้จะเป็น การเปลี่ยนแปลงแบบถาวร มีรูปแบบคือ **BECMG TTGGgg** เมื่อ TTGGgg คือ คาดหมายเวลาที่จะ การเปลี่ยนแปลงจะเริ่มต้นเกิดและเสร็จสมบูรณ์ภายใน 2 ชั่วโมงข้างหน้า โดย TT = ตัวย่อ ดังนี้ FM (From), TL (Until) หรือ AT (At) ตามความเหมาะสม GGgg = เวลาที่คาดหมายเป็นชั่วโมง (GG) และนาทิจ (gg) UTC มีรูปแบบการใช้คาดหมายการเปลี่ยนแปลง ดังนี้

(1) คาดหมายทั้งเวลาที่จะเปลี่ยนแปลงจะเริ่มต้นเกิดและเสร็จสมบูรณ์ ให้ใช้ ตัวย่อ FM และ TL เช่น BECMG FM1230 TL1330

(2) คาดหมายเฉพาะเวลาที่จะเปลี่ยนแปลงจะเสร็จสมบูรณ์ ให้ใช้ตัวย่อ TL

(3) คาดหมายเฉพาะเวลาที่จะเปลี่ยนแปลงจะเริ่มต้นเกิด ให้ใช้ตัวย่อ FM

(4) คาดหมายเวลาเกิดการเปลี่ยนแปลงที่แน่นอน ให้ใช้ตัวย่อ AT

(5) ไม่มีการคาดหมายเวลาการเปลี่ยนแปลง ไม่ต้องใช้ตัวย่อ FM, TL หรือ AT

- TEMPO (มาจาก Temporary) เป็นการเปลี่ยนแปลงแบบครั้งคราว จะใช้เมื่อ คาดหมายว่า สาระประกอบอุตุนิยมวิทยาเปลี่ยนแปลงถึงค่าหรือผ่านค่าที่สำคัญที่กำหนดไว้แบบผัน ผวน (fluctuations) โดยเวลาของเหตุการณ์การเปลี่ยนแปลงที่เกิดขึ้นในแต่ละครั้งน้อยกว่า 1 ชั่วโมง และเมื่อรวมเวลาที่เกิดขึ้นของ เหตุการณ์แต่ละครั้งเข้าด้วยกัน ต้องน้อยกว่าครึ่งหนึ่งของช่วงเวลาการ พยากรณ์ มีรูปแบบคือ **TEMPO TTGGgg** มีรูปแบบการใช้คาดหมายการเปลี่ยนแปลง ดังนี้

(1) คาดหมายทั้งเวลาเริ่มต้นและเวลาสิ้นสุดการเปลี่ยนแปลง ให้ใช้ตัวย่อ FM และ TL เช่น TEMPO FM1230 TL1330

(2) คาดหมายเฉพาะเวลาสิ้นสุดการเปลี่ยนแปลง ให้ใช้ตัวย่อ TL

(3) คาดหมายเฉพาะเวลาเริ่มต้นการเปลี่ยนแปลง ให้ใช้ตัวย่อ FM

(4) ไม่มีการคาดหมายเวลาการเปลี่ยนแปลง ไม่ต้องใช้ตัวย่อ FM หรือ TL

2) ลมผิวพื้น ตามรูปแบบ dddffKT หรือ dddffGf_mf_mKT ซึ่งในการคาดหมายแนวโน้มทิศทางลม จะไม่ใช่ VRB (Variable)

3) ทิศนวิสัย ตามรูปแบบ “VVVV” คือ ค่าทัศนวิสัยที่คาดหมาย ประกอบด้วยตัวเลข 4 ตัว มีหน่วยเป็นเมตร โดยทั่วไป ค่าทัศนวิสัยที่คาดหมายจะเป็นทัศนวิสัยที่เด่นชัด (Prevailing visibility) ของสนามบิน ยกเว้น กรณีคาดหมายการเกิดสภาพอากาศที่มีนัยสำคัญ ที่ทำให้ทัศนวิสัยลดลงจนมีผลกระทบต่อปฏิบัติการด้านการบิน (ทัศนวิสัยต่ำกว่า 5000 เมตร) เช่น พายุฝนฟ้าคะนอง หมอกหนา (Fog) เป็นต้น ให้คาดหมายค่าทัศนวิสัยต่ำสุด (minimum/lowest visibility) และ กรณีที่มีการเปลี่ยนแปลงเกี่ยวกับทัศนวิสัย ต้องระบุสภาพอากาศ ที่ทำให้ทัศนวิสัย มีการเปลี่ยนแปลงด้วย

4) สภาพอากาศ ตามรูปแบบ “w'w'” หรือ NSW โดยที่ w'w' คือ รหัสอักษรย่อสภาพอากาศที่คาดหมาย ประกอบด้วยการพยากรณ์ประเภทและความรุนแรงของสภาพอากาศนั้น ๆ และ NSW คือ ไม่มีหรือสิ้นสุดสภาพอากาศที่มีนัยสำคัญ (Nil Significant Weather)

5) เมฆ หรือ ทัศนวิสัยในแนวตั้ง ตามรูปแบบ “N_sN_sN_{sh}sh_s(CC)” หรือ “VW_{sh}sh_s” หรือ “NSC”

บทที่ 3

การใช้ปัญญาประดิษฐ์ช่วยในการทำเว็บไซต์ และการเผยแพร่เว็บไซต์ทางอินเทอร์เน็ต

3.1 ปัญญาประดิษฐ์

ปัญญาประดิษฐ์ (Artificial Intelligence: AI) คือ โปรแกรมคอมพิวเตอร์ที่มีฟังก์ชันที่สามารถทำงานได้คล้ายกับมนุษย์ และสามารถเลียนแบบการทำการกิจกรรมของมนุษย์ได้ เช่น การเรียนรู้ การวางแผน และการแก้ไขปัญหาต่างๆ เป็นตัวช่วยมนุษย์ในการคิด ซึ่งจะเน้นไปในเรื่องของการประมวลผล และวิเคราะห์ข้อมูลต่างๆ ซึ่งปัจจุบัน AI ได้เข้ามามีบทบาทมากมายในชีวิตประจำวันของเราในหลายๆ ด้าน อาทิ ใช้ในการสื่อสาร การเรียนรู้ หรือแม้กระทั่งการทำงานต่างๆ ที่สามารถช่วยเพิ่มประสิทธิภาพและความสะดวกสบายได้อย่างมาก เช่น การใช้แชทบอท (Chat Bot) ช่วยตอบคำถาม การใช้ระบบแนะนำสินค้าบนเว็บไซต์ หรือแม้แต่การใช้ AI ในการวิเคราะห์ข้อมูลเพื่อช่วยในการตัดสินใจในธุรกิจ การแพทย์ หรือการวิจัยต่างๆ นอกจากนี้ AI ยังมีส่วนสำคัญในการพัฒนานวัตกรรมใหม่ๆ ที่อาจเปลี่ยนแปลงวิถีชีวิตและสังคมในอนาคตอย่างมหาศาล การใช้ AI อย่างถูกต้องและมีจริยธรรมจึงเป็นสิ่งสำคัญในการเตรียมตัวสำหรับการพัฒนาที่เกิดขึ้นอย่างรวดเร็วในยุคนี้

3.1.1 หลักการทำงานของ AI

ระบบ AI ทำงานโดยการรับข้อมูล วิเคราะห์ข้อมูล และประมวลผล เพื่อให้ได้ผลตอบกลับมา ไม่ว่าจะเป็นการใช้คำพูด ข้อความ หรือการกระทำต่างๆ ผลที่ตอบกลับมาก็อยู่ที่ว่าเราต้องการให้ตอบกลับมาเป็นแบบไหน และเอาผลลัพธ์นั้นมาใช้ประโยชน์ให้ตรงกับจุดประสงค์ของเรา อีกทั้งยังสามารถใช้รูปแบบการทำงานนี้เพื่อคาดการณ์สถานการณ์ที่จะเกิดขึ้นต่อในอนาคตได้ เช่น แชทบอทที่ตอบข้อความอัตโนมัติได้เหมือนกับคน หรือความสามารถในการจดจำภาพ ซึ่งการทำงานของระบบทั้งหมดนี้ต้องถูกเขียนโปรแกรมขึ้นมา โดยการเขียนโปรแกรมของ AI นั้นจะเน้นไปที่ทักษะการรับรู้ต่างๆ ดังนี้

1) การเรียนรู้ (Learning) โดยจะเน้นไปที่การรับข้อมูล และสร้างกฎสำหรับการเปลี่ยนเป็นข้อมูลที่น่าไปใช้ได้จริง ซึ่งกฎนั้นเรียกว่า อัลกอริทึม (Algorithms) คือ กระบวนการแก้ปัญหาที่อธิบายเป็นขั้นตอนไว้อย่างชัดเจน

2) การใช้เหตุผล (Reasoning) เน้นการตัดสินใจเลือกอัลกอริทึมที่เหมาะสมเพื่อให้ได้ผลลัพธ์ที่ต้องการ

3) การแก้ไขข้อผิดพลาด (Self-correction) ในส่วนนี้จะได้รับการออกแบบเพื่อปรับแต่งอัลกอริทึมให้วิเคราะห์ได้อย่างละเอียด เพื่อรับประกันว่าจะได้ผลลัพธ์ที่แม่นยำที่สุด

4) การมีความคิดสร้างสรรค์ (Creativity) เป็นส่วนที่ใช้เครือข่ายประสาทเทียมอิงตามกฎ วิธีทางสถิติ และเทคนิคอื่นๆ เพื่อให้สามารถสร้างภาพใหม่ เพลงใหม่ หรือแนวคิดใหม่ๆ ได้

3.1.2 ประเภทของปัญญาประดิษฐ์

ปัญญาประดิษฐ์แบ่งตามความสามารถ

1) Artificial Narrow Intelligence (ANI) หรือ Weak AI เป็นระบบ AI ประเภทที่สามารถทำงานแทนมนุษย์ได้เพียงบางอย่างเท่านั้น และจำกัดอยู่แค่ในวงแคบ คือต้องเป็นงานหรือทักษะที่ได้รับการโปรแกรมชุดคำสั่งมาเท่านั้น ไม่สามารถทำงานนอกเหนือจากนั้นได้ และไม่สามารถเรียนรู้จากขอบเขตที่ตั้งไว้ได้ หรือ เรียนรู้ได้เฉพาะภายในงานแคบ ๆ เท่านั้น ซึ่งนั่นคือ AI ที่เราใช้กันอยู่ในทุกวันนี้ เช่น หุ่นยนต์บริการที่เห็นได้ในร้านอาหารต่างๆ

2) Artificial General Intelligence (AGI) หรือ Strong AI เป็นระบบ AI ประเภทที่มีสติปัญญา และความสามารถในการทำงานต่างๆ ได้เทียบเท่ากับสมองมนุษย์สามารถคิดวิเคราะห์ วางแผน และสามารถแก้ปัญหาต่างๆ ได้ เข้าใจเรื่องที่เป็นนามธรรม นอกจากนี้ ยังสามารถเรียนรู้ประสบการณ์จากอดีตได้เหมือนมนุษย์ แต่ในปัจจุบันยังไม่สามารถพัฒนา AI ประเภทนี้ได้สำเร็จ ซึ่ง AI ประเภทนี้อาจเป็นภัยต่อการดำรงอยู่ของมนุษย์ได้เช่นกัน

3) Artificial Super Intelligence (ASI) เป็นระบบ AI ประเภทที่มีปัญญามากกว่ามนุษย์ ซึ่งปัจจุบันยังไม่สามารถพัฒนา AI ในระดับนี้ขึ้นมาได้ ส่วนมากจะเป็นไอดีที่เกิดขึ้นในสื่อต่างๆ มากมาย เช่น ภาพยนตร์ ซีรีส์ นวนิยาย หรือเกม โดยไอดีที่เกิดขึ้นเป็นแนวคิดเรื่องเครื่องจักรสามารถยึดครองโลกได้

ปัญญาประดิษฐ์แบ่งตามระบบการประมวลผล

1) Reactive Machines เป็น AI ขั้นพื้นฐานที่ตัดสินใจด้วยการนำข้อมูลเฉพาะหน้ามาประมวลผลเท่านั้น AI ประเภทนี้ไม่มีหน่วยความจำหรือข้อมูลย้อนหลังเพื่อใช้ตัดสินใจ ทำให้การทำงานของ Reactive Machines เป็นการตัดสินใจแบบ case by case เช่น การให้ตัวเลือกที่แนะนำใน Netflix ถ้าเทียบกับมนุษย์ก็เหมือนกับการตัดสินใจด้วยสัญชาตญาณนั่นเอง

2) Limited Memory เป็น AI ขั้นถัดมาที่จำและเก็บข้อมูลได้บ้าง ประสิทธิภาพการตัดสินใจของ Limited Memory ขึ้นอยู่กับปริมาณข้อมูลที่ถูกป้อนเข้าไป AI ประเภทนี้มีวิธีการเรียนรู้เป็นของตัวเอง ทั้งนี้ AI ประเภทนี้ไม่สามารถเรียนรู้ผ่านประสบการณ์ได้ และเรียนรู้ได้เฉพาะข้อมูลที่ถูกป้อนเข้าไปเท่านั้น ถ้าเทียบกับมนุษย์ AI ประเภทนี้ก็เหมือนกับการเรียนรู้ผ่านการท่องจำ

3) Theory of Mind เป็น AI ขั้นถัดมาที่กำลังอยู่ในขั้นพัฒนา AI ประเภทนี้ต่างกับ 2 ประเภทแรกที่มี “การมีจิตจิตใจ” เพื่อให้ AI ประเภทนี้ตัดสินใจอย่างมีวิจารณญาณและเข้าใจอารมณ์ของมนุษย์ เช่น หาก AI เข้าใจหรือรับรู้ได้ว่ามนุษย์หิว AI อาจนำของว่างมาให้ หากมนุษย์เรียนรู้สังคมด้วยการเข้าใจอารมณ์ของตนเองและผู้อื่น AI ประเภทนี้อาจเข้าใจสังคมของมนุษย์ด้วยการเข้าใจอารมณ์ของมนุษย์ได้ด้วยเช่นกัน

4) Self – awareness เป็น AI ขั้นสูงสุด สามารถมีอารมณ์ ความรู้สึก ความเชื่อ และมีความต้องการเป็นของตัวเอง รวมถึงสามารถคิดตัดสินใจ เลือก และกระทำการต่างๆ ได้ด้วยตนเองทั้งหมด ซึ่งในปัจจุบันก็ยังไม่สามารถพัฒนาขึ้นมาได้เช่นเดียวกับประเภท Theory of Mind ซึ่ง AI ประเภทนี้มีความก้าวกระโดดกว่าทฤษฎีความคิดของ AI เป็นอย่างมาก ตั้งแต่การทำความเข้าใจอารมณ์ ไปสู่การตระหนักรู้ในสถานะของตนเอง การคาดการณ์ความรู้สึกของผู้อื่นได้ และมนุษย์เรายังห่างไกลจาก AI ประเภทนี้มาก

3.1.3 Generative Artificial Intelligence (Generative AI)

ปัจจุบันได้มีการพัฒนา AI ขึ้นมาหลายประเภท ซึ่งมีความสามารถและลักษณะเด่นที่แตกต่างกันไป บางประเภทสามารถสร้างเนื้อหาหรือข้อมูลใหม่จากการเรียนรู้และวิเคราะห์ข้อมูลที่มีอยู่แล้ว ไม่เพียงแค่ประมวลผลข้อมูล แต่ยังสามารถสร้างสิ่งใหม่ที่ไม่เคยมีมาก่อนได้ เรียกว่า Generative AI (Gen-AI) จัดอยู่ใน AI ประเภท ANI และ Limited Memory โดยสามารถนำมาใช้งานได้หลายด้าน เช่น การสร้างข้อความ การสร้างภาพ การสร้างเสียงหรือดนตรี และการสร้างวิดีโอ ซึ่งสามารถช่วยสนับสนุนการทำงาน การเรียนรู้ การแก้ปัญหา หรือแม้กระทั่งช่วยเพิ่มประสิทธิภาพในชีวิตประจำวัน การวิเคราะห์ข้อมูลและการคาดการณ์แนวโน้มตลาด ทำให้ธุรกิจสามารถตัดสินใจได้อย่างมีประสิทธิภาพยิ่งขึ้น ตัวอย่าง Generative AI ในด้านต่างๆ ดังภาพที่ 3.1



ภาพที่ 3.1 แสดงตัวอย่าง Generative AI ในด้านต่างๆ

ที่มา: <https://dataholicth.com/info/generative-ai-landscape>

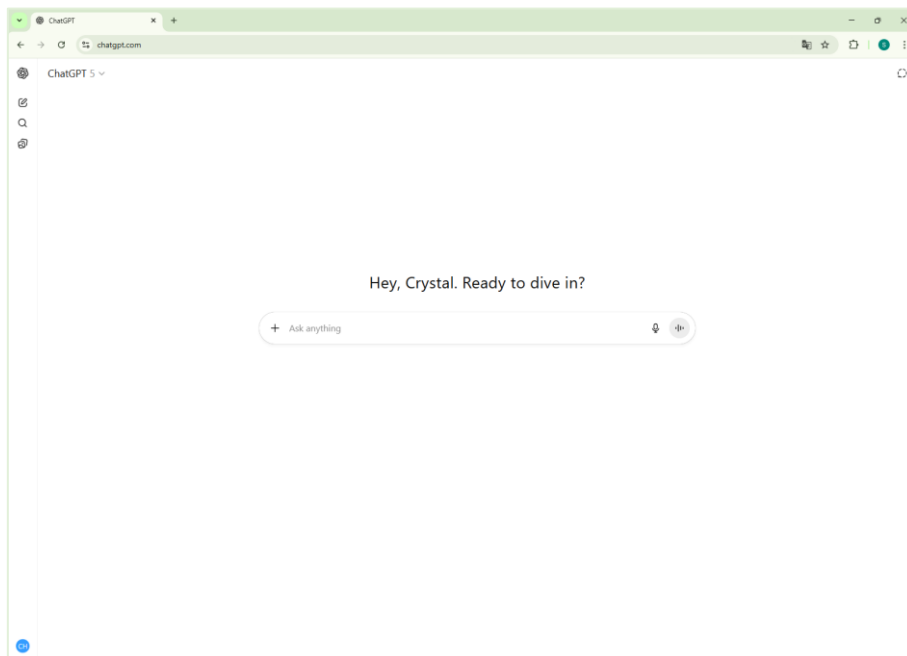
3.2 การใช้ปัญญาประดิษฐ์ช่วยในการทำเว็บไซต์

การทำเว็บไซต์ อาจจะมองว่ายากหากไม่มีพื้นฐานทางด้านโปรแกรม แต่ถ้าหากมีความสนใจในการทำเว็บไซต์ การนำ AI เข้ามาช่วย ก็อาจเป็นสิ่งที่สามารถทำได้ เพราะ AI สามารถแนะนำเครื่องมือในการทำเว็บไซต์ แม้แต่สร้างโค้ดให้เลย ก็สามารถทำได้ เพียงเรารู้ว่าเรามีข้อมูลนำเข้าอะไร แล้วเราอยากได้ผลลัพธ์ออกมาอย่างไร ให้เราคิดกระบวนการทำงาน แล้วให้ AI ช่วยจัดการกระบวนการย่อยเหล่านั้น แล้วเราก็เอามาประกอบเข้าด้วยกัน พร้อมกับศึกษาสิ่งที่ AI ให้เรามา

Generative AI ด้านการสร้างข้อความ ที่สามารถช่วยในการสร้างโค้ดภาษาคอมพิวเตอร์ ที่เป็นที่นิยมในปัจจุบัน เช่น ChatGPT ของ OpenAI, Claude ของ Anthropic, Microsoft Copilot ของ Microsoft, Gemini ของ Google ในครั้งนี้จะใช้ ChatGPT เป็นหลัก เพราะสามารถตอบโต้และดำเนินเรื่องราวต่อเนื่องได้ดี และสามารถใช้งานฟรี (แบบมีข้อจำกัด) แต่ถ้าผลลัพธ์ไม่เป็นที่น่าพอใจ ก็สามารถใช้ AI ตัวอื่นช่วยได้ เพราะบางครั้ง ฐานข้อมูลที่ AI มี ก็อาจเป็นข้อมูลเก่าและยังไม่ได้อัปเดต ทำให้แสดงข้อมูลที่ล้าหลัง และไม่ถูกต้องในปัจจุบันได้ ซึ่งเราสามารถชำระเงินเพื่อใช้งาน ChatGPT ที่มีประสิทธิภาพสูงกว่าแบบใช้งานฟรีได้

3.3 ตัวอย่างการใช้ ChatGPT ในการทำเว็บไซต์

1) เข้าไปยังเว็บไซต์ <https://chatgpt.com>



ภาพที่ 3.2 หน้าเว็บไซต์ <https://chatgpt.com>

2) พิมพ์ข้อความหรือคำถาม ที่ต้องการให้ AI ช่วย หรือที่เรียกว่า พรอมต์ (Prompt) โดยต้องใช้คำที่ชัดเจนและเฉพาะเจาะจง ให้ข้อมูลเพิ่มเติมที่จำเป็น และกำหนดรูปแบบของผลลัพธ์ที่ต้องการ แต่บางครั้ง แม้เราจะพิมพ์พรอมต์ที่กำกวม AI ก็ฉลาดพอที่จะตีความได้ แต่การพิมพ์พรอมต์ที่ชัดเจน จะให้ผลลัพธ์ที่ดีกว่า ตัวอย่างเช่น ต้องการแสดงผลการตรวจอากาศจากสถานีอุตุนิยมวิทยา

url: <https://www.tmd.go.th/api/xml/weather-report?stationnumber=48583> มีรูปแบบข้อมูลคือ

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<rss xmlns:content="http://purl.org/rss/1.0/modules/content/"
xmlns:dc="http://purl.org/dc/elements/1.1/" version="2.0">
<channel>
<title>รายงานสภาวะอากาศ - นราธิวาส : อ.เมือง จ.นราธิวาส</title>
<link>http://www.tmd.go.th</link>
<description>รายงานสภาวะอากาศ โดย กรมอุตุนิยมวิทยา กระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม</description>
<language>th-TH</language>
<copyright>Copyright 2006,Thai Meteorological Deaprtment</copyright>
<ttl>20</ttl><item>
<title>รายงานสภาวะอากาศ - นราธิวาส : อ.เมือง จ.นราธิวาส วันที่ : 14/08/2025 เวลา 13:00 นาฬิกา</title>
<pubDate>Thu, 14 Aug 2025 13:00:00 +07:00</pubDate>
<guid>48583</guid>
<link>http://www.tmd.go.th/api/xml/weather-report?stationnumber=48583</link>
<author>กรมอุตุนิยมวิทยา</author>
<description>
<![CDATA[ <b> อุณหภูมิ : </b>33.0 องศาเซลเซียส <br/><b>ความชื้นสัมพัทธ์ : </b> 55 %
<br/><b>ความกดอากาศ : </b> 1007.94 มิลลิบาร์ <br/><b>ทิศทางลม : </b> ทิศตะวันตก <b>
ความเร็ว </b>7.4 กม./ชม.<br/><b>ทัศนวิสัย : </b>10.0 กิโลเมตร <br/><b>ลักษณะอากาศ : </b>
มีเมฆเป็นส่วนมาก<br/><b>ฝนสะสมวันนี้ : </b> 0.0 มิลลิเมตร<br/><b>พระอาทิตย์ขึ้นเข้าพรั่งนี้:
<br/><b>06:07 น. <br/><b>พระอาทิตย์ตกเย็นวันนี้: <br/>18:27 น. ]]>
</description></item></channel></rss>
```

ขอ html แบบเต็ม ให้โหลดข้อมูลผ่าน proxy

[https://corsproxy.vtsc48583.workers.dev/?url=\\${encodeURIComponent\(url\)}](https://corsproxy.vtsc48583.workers.dev/?url=${encodeURIComponent(url)})

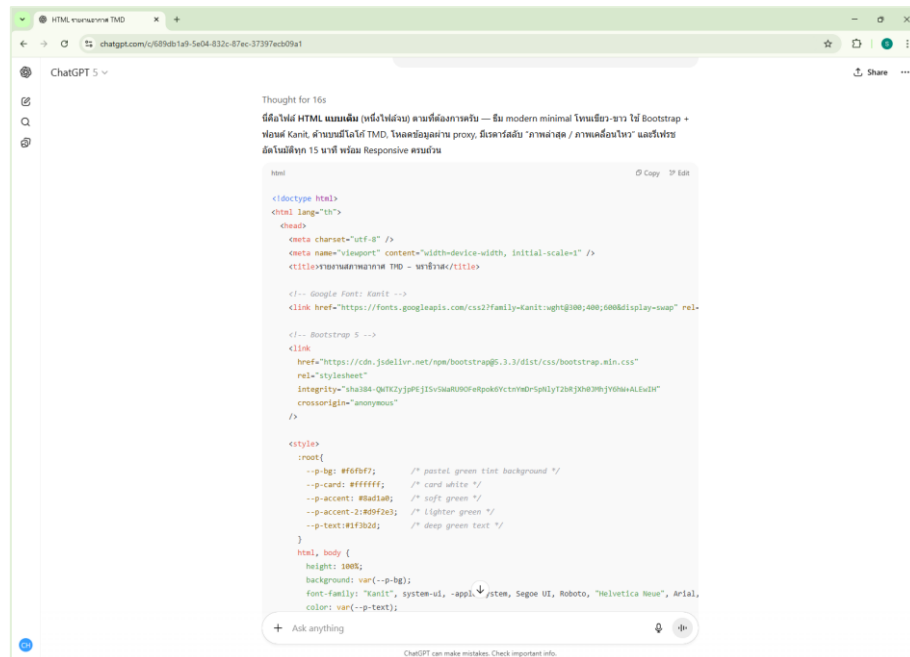
- ขอรูปแบบ modern minimal pastel อิมสีเขียว-ขาว สวยงาม มี emoji ประกอบ และมีโลโก้

https://www.tmd.go.th/media/ServicePages/Identities/Logoes/logo_colorcmymk.png อยู่ด้านบน

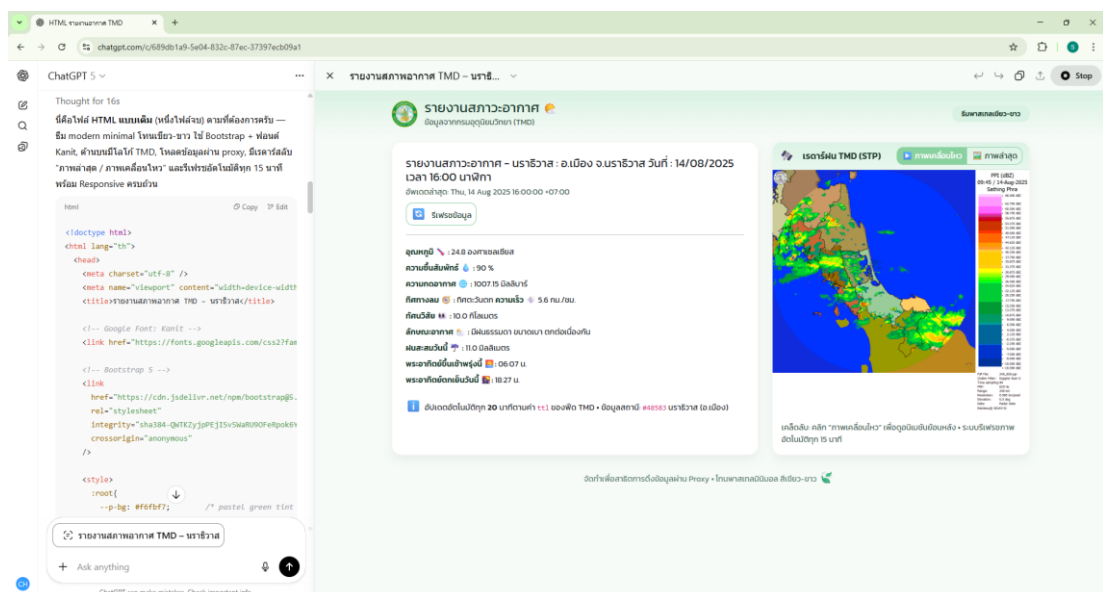
- ใช้ฟอนต์ Kanit

- ใช้ bootstrap ตกแต่ง และทำ Responsive Web Design

- ให้แสดง https://weather.tmd.go.th/stp/stp240_latest.gif อยู่ด้านข้าง (ถ้าหน้าจอบแคบ ให้ย้ายไปด้านล่าง) ถ้าหากคลิกปุ่ม ให้เปลี่ยนเป็น <https://weather.tmd.go.th/stp/stp240Loop.gif> ให้อัปเดตภาพเรดาร์ทุกๆ 15 นาที



ภาพที่ 3.3 แสดงโค้ดภาษา HTML พร้อมกับคำอธิบายที่ ChatGPT สร้างขึ้นมา

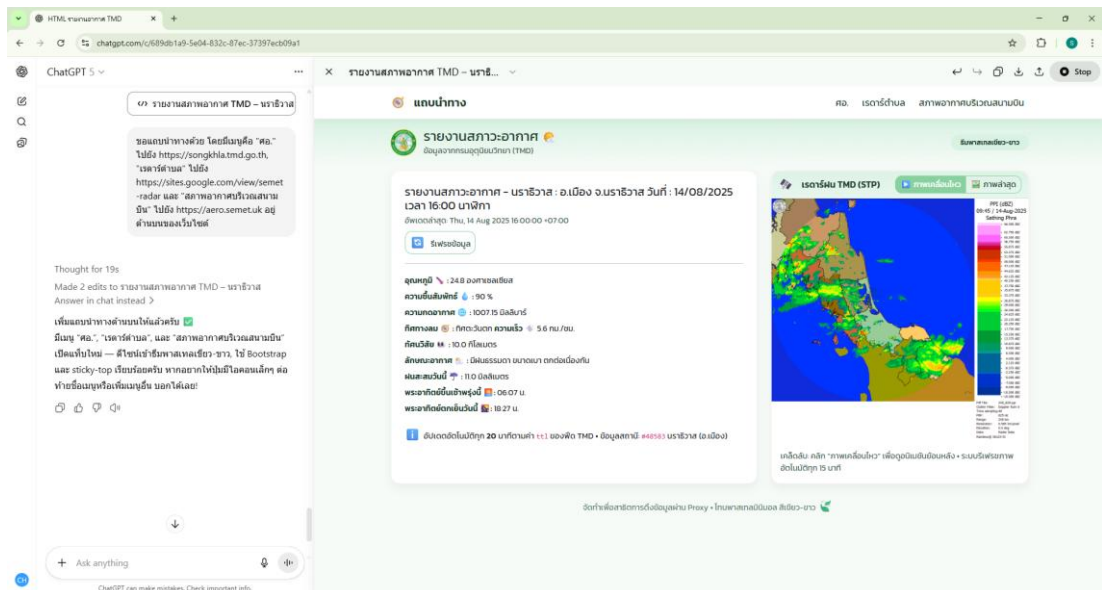


ภาพที่ 3.4 แสดงผลลัพธ์ของโค้ดภาษา HTML ที่ ChatGPT สร้างขึ้นมา

3) ทดสอบพิมพ์ฟรอมต์เพิ่มเติม

“ขอแนะนำทางด้วย โดยมีเมนูคือ "ศอ." ไปยัง <https://songkhla.tmd.go.th>, "เรดาร์ตำบล" ไปยัง <https://sites.google.com/view/semet-radar> และ "สภาพอากาศบริเวณสนามบิน" ไปยัง <https://aero.semet.uk> อยู่ด้านบนของเว็บไซต์”

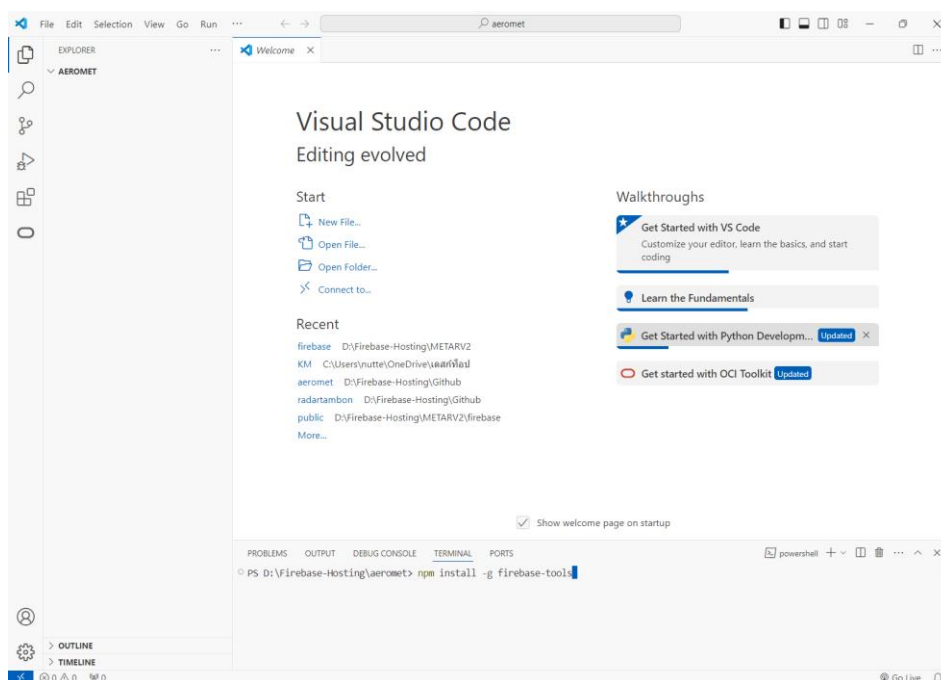
จะได้ผลลัพธ์คือ เพิ่มแถบนำทาง (Navbar) ตามที่ต้องการ และสามารถไปยังลิงก์ต่างๆได้



ภาพที่ 3.5 แสดงผลลัพธ์เพิ่มเติมที่ ChatGPT สร้างขึ้นมา

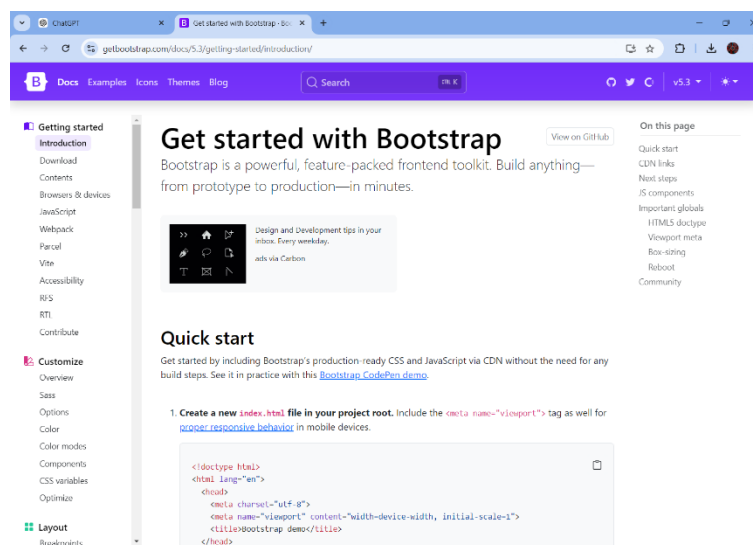
3.4 แนะนำเครื่องมือเพิ่มเติมสำหรับทำเว็บไซต์

3.4.1 Visual Studio Code หรือ VS Code คือโปรแกรมแก้ไขโค้ด (Code Editor) ที่พัฒนาโดย Microsoft ได้รับความนิยมอย่างมากในหมู่นักพัฒนา เพราะมีคุณสมบัติและฟีเจอร์ที่หลากหลาย รองรับได้หลายภาษา และยังสามารถปรับแต่งการทำงานผ่านส่วนขยาย (Extensions) ที่มีให้เลือกมากมาย ทั้งยังสามารถตรวจสอบโค้ดที่ไม่ถูกต้องได้ด้วย และสำหรับการพัฒนาเว็บไซต์นั้น ยังมีส่วนขยายที่สามารถทำเซิร์ฟเวอร์เทียมได้ เช่น Live Server ทำให้ทดสอบเว็บไซต์ได้ง่าย ผ่านการแสดงผลแบบเรียลไทม์ และสามารถแสดงผลบนอุปกรณ์อื่นผ่าน Local Network ได้ด้วย ดาวน์โหลดโปรแกรมได้ที่ <https://code.visualstudio.com>



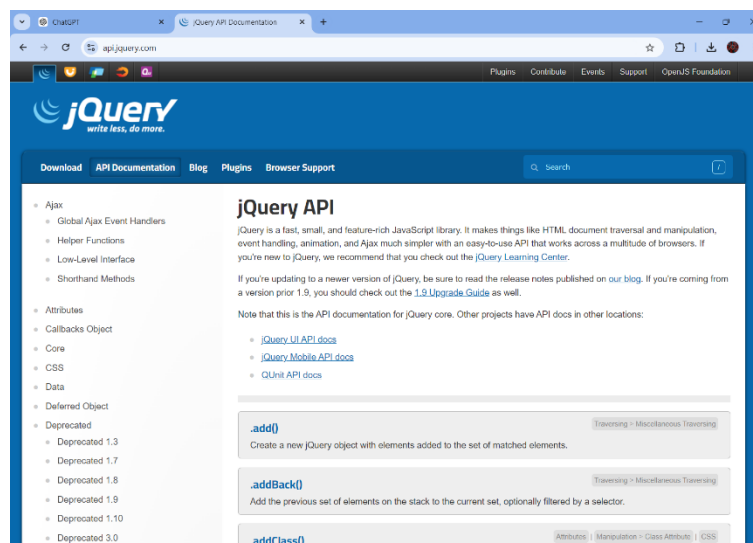
ภาพที่ 3.6 แสดงโปรแกรม Visual Studio Code

3.4.2 Bootstrap คือเครื่องมือสำหรับการพัฒนาเว็บที่ช่วยให้การสร้างเว็บไซต์ ที่สามารถปรับขนาดตามอุปกรณ์ (Responsive) ให้ง่ายและรวดเร็วขึ้น โดยมีชุดเครื่องมือที่รวม CSS, JavaScript และคอมโพเนนต์ UI ที่ออกแบบมาให้ใช้งานง่าย เช่น ปุ่ม, แบบฟอร์ม, และแถบการนำทาง นอกจากนี้ยังรองรับการปรับแต่งและใช้งานได้บนหลายแพลตฟอร์ม เข้าไปดูรายละเอียดและวิธีการใช้งานได้ที่ <https://getbootstrap.com/docs/5.3/getting-started/introduction>



ภาพที่ 3.7 แสดงเว็บไซต์ <https://getbootstrap.com>

3.4.3 jQuery คือไลบรารี JavaScript ที่ทำให้การเขียนโค้ด JavaScript ง่ายขึ้น โดยช่วยให้สามารถจัดการกับองค์ประกอบในหน้าเว็บ ทำงานกับ DOM (Document Object Model), จัดการเหตุการณ์ (events), และทำเอฟเฟกต์ต่างๆ ได้ง่ายขึ้น jQuery ใช้สั้นและชัดเจน ทำให้การพัฒนาเว็บรวดเร็วและสะดวกมากขึ้น ดูรายละเอียดและวิธีการใช้งานได้ที่ <https://api.jquery.com>



ภาพที่ 3.8 แสดงเว็บไซต์ <https://api.jquery.com>

3.5.1 ภาษา HTML

ภาษา HTML ย่อมาจาก Hypertext Markup Language เป็นภาษาพื้นฐานที่ใช้สำหรับสร้างเว็บเพจ จะแสดงผลผ่านทางเว็บเบราว์เซอร์ ส่วนเว็บเบราว์เซอร์ที่นิยมเช่น Google Chrome, Firefox, Safari, Microsoft Edge เป็นต้น เบราว์เซอร์จะมีหน้าที่สำหรับรับไฟล์ HTML ขึ้นมาหรือรับหน้าเว็บที่เราเขียนขึ้นมา โดยจะแสดงข้อมูลตัวอักษร ภาพ วิดีโอ เสียง และไฟล์ในรูปแบบอื่นๆ มีการเชื่อมโยงลิงก์ของแต่ละเพจด้วย โดย HTML มีโครงสร้างภายในประกอบไปด้วยสิ่งที่เรียกว่าแท็ก (Tag) โดยแท็ก จะเป็นชุดคำสั่งเพื่อให้เกิดการแสดงผลในรูปแบบต่าง ๆ และในแต่ละแท็ก จะมีส่วนที่เรียกว่าแอททริบิวต์ (Attribute) เพื่อควบคุมการทำงานของแท็กแต่ละตัว ซึ่งตัวไฟล์ HTML จะมีนามสกุล .html สามารถเพิ่ม แท็ก <script> ไว้สำหรับรัน JavaScript และ แท็ก <style> ไว้สำหรับปรับแต่ง CSS ได้เลย หรือจะแยก JavaScript และ CSS ออกมาเป็นไฟล์ไว้ข้างนอกก็ได้

โครงสร้างของภาษา HTML

```
<!DOCTYPE html>
<html lang="th">
  <head>
    .....
  </head>
  <body>
    .....
  </body>
</html>
```

ส่วน head เป็นส่วนที่อยู่ภายในแท็ก <head>...</head> ใช้สำหรับอธิบายข้อมูลเกี่ยวกับเว็บ เช่น ชื่อเรื่องของเว็บไซต์ คำอธิบาย คีย์เวิร์ดสำหรับสืบค้น ไอคอน รวมถึงเรียกใช้ JavaScript และ CSS ก่อนจะโหลดส่วนเนื้อหา เป็นต้น

ส่วน body เป็นส่วนที่อยู่ภายในแท็ก <body>...</body> ใช้อธิบายเนื้อหาหลักของเว็บไซต์ เช่น ใส่ข้อความต่างๆ รูปภาพ แบบฟอร์ม วิดีโอ และยังสามารถกำหนดคุณสมบัติพื้นฐานของเว็บได้ เช่น รูปแบบของพื้นหลัง สีของตัวอักษร

ตัวอย่างแท็ก

```
<!DOCTYPE html>
<html lang="th">
  <head>
    <title>ชื่อเว็บ</title>
    <link rel="stylesheet" href="styles.css">
    <style>ภาษา CSS</style>
  </head>
  <body>
    <h2>หัวข้อต่าง ๆ</h2>
    <p>เนื้อหา</p>
    
    <script src="scripts.js"></script>
    <script>ภาษา JavaScript</script>
  </body>
</html>
```

3.5.2 ภาษา CSS

ภาษา CSS หรือ Cascading style sheet นั้น เป็นภาษาเขียนเว็บที่มักจะใช้คู่กับ HTML โดยมีหน้าที่ในการกำหนดหน้าตาของ หรือเลย์เอาต์ต่าง ๆ ของเว็บไซต์ รวมไปถึงดูแลและจัดการองค์ประกอบทุก ๆ อย่าง เช่น ขนาดตัวหนังสือ รูปแบบการวาง ขนาด ระยะห่าง ตำแหน่งของ องค์ประกอบ (Element) บนหน้าเว็บให้เกิดความสวยงาม สามารถรวมภาษา CSS ไว้ในไฟล์ HTML ได้ โดยวางไว้ในแท็ก <style>...</style> หรือจะเอาไว้ภายนอกไฟล์ HTML ก็ได้ ซึ่งตัวไฟล์ CSS จะมีนามสกุล .css

3.5.2.1 โครงสร้างของ CSS

```
selector {
  property1: value1;
  property2: value2;
  .....
}
```

selector เช่น Tag{} หรือ #id{} หรือ .class{} หรือ Tag.class{} หรือผสม Tag, #id, .class{} เป็นต้น

3.5.2.2 ตัวอย่าง property และ value

```
.container {
  font-weight: 500; กำหนดความหนาตัวอักษร
  text-align: center; กำหนดตำแหน่งตัวอักษร
  color: red; กำหนดสีตัวอักษร
  font-family: Sarabun; กำหนดรูปแบบตัวอักษร
  background-color: yellow; ใส่สีพื้นหลัง
}
```

3.5.2.3 รูปแบบการเขียน CSS ใน HTML ทำได้ 3 วิธีด้วยกัน คือ

1) Inline โดยใช้ Attribute style ใน HTML Element เช่น

```
<h1 style="color: red; margin: 0.8rem; font-weight: bold;">Header</h1>
```

2) Internal โดยใช้ Element <style> เช่น

```
<head>
  <style>
    body {background-color: powderblue;}
    #id {color: blue;}
    .class {color: red; border: 1px solid #ccc; width: 100%;}
  </style>
</head>
```

3) External โดยใช้ไฟล์ CSS (.css) แล้วลิงก์ไฟล์เข้ามาใน HTML วิธีนี้เป็นวิธีที่นิยมมากที่สุด เนื่องจากสามารถใช้ตกแต่งหน้าเว็บหลายๆ หน้าได้ และการแก้ไข External CSS เพียงไฟล์เดียว จะส่งผลให้หน้าเว็บทั้งเว็บไซต์ซึ่งอาจมีหลายร้อยหน้าเปลี่ยนได้

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

3.5.3 ภาษา JavaScript

ภาษา JavaScript เป็นภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาเว็บไซต์ร่วมกับ HTML เพื่อให้เว็บมีลักษณะแบบไดนามิก คือเว็บสามารถตอบสนองกับผู้ใช้งาน หรือแสดงเนื้อหาที่แตกต่างกันไป โดยจะอ้างอิงตามเว็บเบราว์เซอร์ที่ผู้เข้าชมเว็บใช้งานอยู่ เป็นภาษาที่ทำงานฝั่งผู้ใช้ (Client Side Script) โดยเว็บเบราว์เซอร์จะทำหน้าที่ประมวลผลคำสั่งที่ถูกเขียนขึ้นมาและตอบสนองต่อผู้ใช้ได้ทันที เช่น การแสดงข้อความแจ้งเตือน การตรวจสอบข้อมูลที่ผู้ใช้ป้อน เป็นต้น จึงเป็นภาษาที่ซับซ้อนกว่า HTML และ CSS

3.5.3.1 รูปแบบการเขียน JavaScript ใน HTML ทำได้ 2 วิธีด้วยกัน คือ

1) Internal โดยใช้ Element `<script>` ในส่วนของ `<head>` หรือ `<body>` แต่นิยมวางไว้ท้ายสุดของ `<body>` เนื่องจากบางครั้งต้องรอโหลด DOM ใน HTML ให้เสร็จก่อน ถ้าวางไว้ก่อน อาจทำให้คำสั่งหา Element ใน DOM ไม่เจอ แล้วจะเกิดข้อผิดพลาดได้

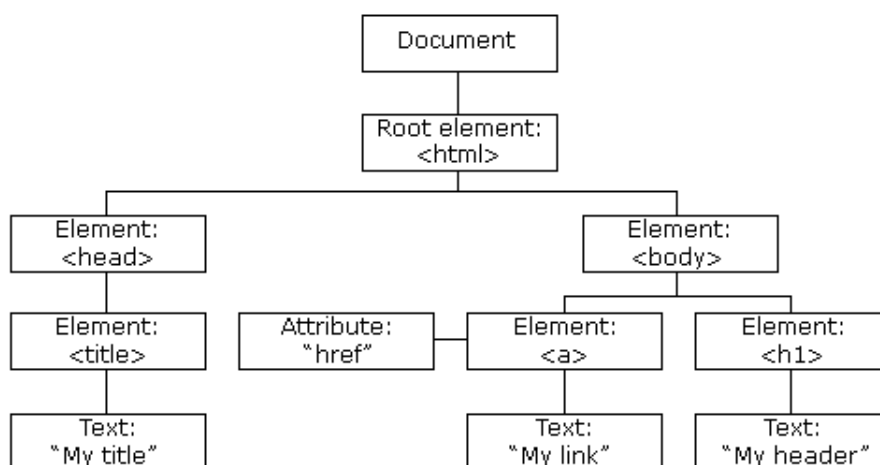
```
<body>
  <script>
    document.addEventListener('DOMContentLoaded', () => {
      const img = document.getElementById("Image");
      img.src = "https://weather.tmd.go.th/stp/stp240Loop.gif";
    });
  </script>
</body>
```

2) External โดยใช้ไฟล์ JavaScript (.js) แล้วลิงก์ไฟล์เข้ามาใน HTML

```
<body>
  <script src="scripts.js"></script >
</body>
```

3.5.3.2 HTML DOM (Document Object Model)

HTML DOM เป็นมาตรฐานที่กำหนดการแสดงผลและการเข้าถึงเอกสารบนเว็บไซต์ เมื่อหน้าเว็บโหลดเสร็จเรียบร้อยแล้ว เว็บเบราว์เซอร์จะสร้าง DOM ของหน้านั้นขึ้นมา ตัว DOM จะทำให้โครงสร้างของเอกสารถูกแปลงเป็นโมเดลข้อมูลแบบต้นไม้ (Tree Structure) ดังภาพที่ 3.10 ซึ่งทำให้ JavaScript สามารถเข้าถึงและปรับเปลี่ยนเอกสาร HTML ได้อย่างง่ายดาย



ภาพที่ 3.10 แสดงโครงสร้างของ HTML DOM

ที่มา: https://www.w3schools.com/js/js_htmlDOM.asp

คุณสมบัติของ HTML DOM คือ เข้าถึงและเปลี่ยนคุณสมบัติทั้งหมดของ Element ในหน้าเว็บได้ ควบคุมและเปลี่ยนรูปแบบ CSS ได้ และสามารถตอบสนองกับทุกเหตุการณ์ที่เกิดขึ้นหน้าเว็บได้

3.5.3.3 การประกาศตัวแปรใน JavaScript

การประกาศตัวแปรใน JavaScript สามารถทำได้ด้วยสามวิธีหลัก ได้แก่ var, let, และ const ซึ่งแต่ละวิธีมีลักษณะและการใช้งานที่แตกต่างกัน โดยใน JavaScript มีการจัดการขอบเขต (Scope) ของตัวแปรและฟังก์ชันเป็น 3 ระดับ ได้แก่ Global Scope, Function Scope และ Block Scope เรียงลำดับความสามารถในการเข้าถึงตัวแปรได้คือ Global Scope > Function Scope > Block Scope นั่นคือ Global Scope สามารถเข้าถึงได้จากทุกที่ และ Block Scope จะเข้าถึงได้เฉพาะในบล็อก ({}) นั้นๆ

```
const x = 10, y = 20;
function difference(a, b){
  if (a > b) {
    let result1 = a - b;
    return result1;
  }
  else {
    var result2 = b - a;
  }
  return result2;
}
console.log(difference(x, y));
```

Block Scope

Block Scope

Function Scope

Global Scope

1) var มีขอบเขตการทำงานในระดับ Function Scope แม้ว่าจะประกาศในบล็อกก็สามารถทะลุบล็อกออกมาได้ ดังตัวอย่างโค้ดด้านบน var result2 ถูกประกาศในบล็อก else ก็สามารถส่งค่าออกนอกบล็อก else ได้ มีข้อดีคือสามารถนำไปใช้ได้ภายในฟังก์ชันนั้นได้ทั้งหมด แต่ก็มีข้อเสียคือ อาจมีการใช้ค่าตัวแปรที่ผิดพลาดได้

2) let มาจากการอัปเดต ES6 (มาตรฐานสำหรับภาษา JavaScript ECMAScript V.6 2015) เพื่อแก้ปัญหาตัวแปร var ทะลุบล็อก มีขอบเขตการทำงานในระดับ Block Scope ทำให้ไม่สามารถเข้าถึงตัวแปรได้จากภายนอกบล็อก แต่ถ้าประกาศตัวแปร let ไว้ที่ Global ก็สามารถเข้าถึงได้จากทุกที่เช่นเดียวกัน หรือถ้าประกาศไว้ที่ฟังก์ชันก็สามารถเข้าถึงได้จากทุกที่ภายในขอบเขตของฟังก์ชันได้ แต่ต้องประกาศตัวแปรก่อนเรียกใช้งาน

3) const มาจากการอัปเดต ES6 เช่นเดียวกันกับ let มีขอบเขตการทำงานในระดับ Block Scope เช่นเดียวกัน แต่วกกันที่ const ใช้สำหรับประกาศค่าตัวแปรที่ไม่ต้องการให้เปลี่ยนแปลงค่าได้ แต่ยังคงเปลี่ยนแปลงคุณสมบัติในตัวแปรเชิงวัตถุ (Object) หรือสมาชิกของตัวแปรอาร์เรย์ (Array) ได้

3.6 การนำเว็บไซต์ไปเผยแพร่ (Website Deployment) ผ่านผู้ให้บริการ Hosting

สำหรับนักพัฒนาที่ไม่มีเซิร์ฟเวอร์เป็นของตัวเอง การเลือกใช้บริการ Hosting จากผู้ให้บริการที่มีอยู่ ถือเป็นทางเลือกที่ดี เนื่องจากการจัดตั้งและดูแลเซิร์ฟเวอร์เองนั้นมีค่าใช้จ่ายสูง มีค่าใช้จ่ายทั้งเรื่องการเช่าไอพีแอดเดรส จดทะเบียนชื่อโดเมน และการรักษาความปลอดภัยของเซิร์ฟเวอร์ รวมทั้งต้องเปิดเครื่องให้ทำงานตลอด 24 ชั่วโมง เพื่อให้เว็บไซต์สามารถเข้าถึงได้ตลอดเวลา ซึ่งการใช้บริการ Hosting มีทั้งแบบที่เสียค่าใช้จ่ายรายเดือนรายปี หรือตามปริมาณการใช้งานจริง นอกจากนี้ ยังมีแบบฟรีให้ได้ใช้งาน แต่ก็มีข้อจำกัดบางประการ เช่น

- 1) มีการจำกัดระยะเวลาที่เซิร์ฟเวอร์ใช้ในการสร้าง (Build) โปรเจกต์ คือเวลาที่ใช้ในการคอมไพล์โค้ด รวมถึงไลบรารี และปลั๊กอินที่โปรเจกต์ใช้ ให้สามารถใช้งานบนอินเทอร์เน็ตได้
- 2) จำกัดปริมาณข้อมูลที่ส่งผ่านโฮสติ้ง เรียกว่า แบนด์วิดท์ (Bandwidth)
- 3) จำกัดพื้นที่จัดเก็บข้อมูล
- 4) จำกัดจำนวนการเข้าถึง หรือจำนวนการเข้าใช้งานพร้อมกัน
- 5) ชื่อโดเมนที่ถูกกำหนดโดยผู้ให้บริการโฮสติ้ง อาจจะไม่เป็นที่น่าพอใจ (ถ้าหากมีโดเมนอยู่แล้วก็สามารถเอามาเปลี่ยนได้)
- 6) อาจไม่ได้รับการช่วยเหลือจากผู้ให้บริการอย่างเต็มที่
- 7) อาจมีโฆษณาจากผู้ให้บริการแสดงบนเว็บไซต์ของเรา

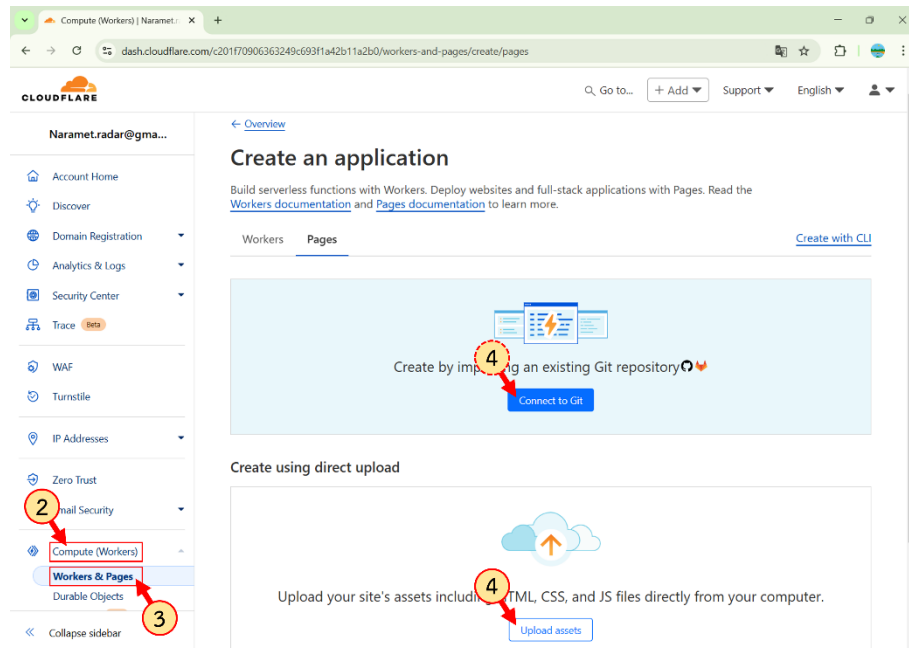
ในหัวข้อนี้จะแนะนำผู้ให้บริการโฮสติ้ง (Hosting) แบบฟรี หากเป็นเว็บไซต์ที่มีขนาดเล็ก การประมวลผลไม่ได้ซับซ้อน และใช้งานเพื่อแสดงผลเฉพาะทาง เมื่อพิจารณาขนาดของโปรเจกต์เว็บไซต์และการจัดการแล้ว โฮสติ้งแบบฟรี ก็เพียงพอต่อการใช้งาน การจัดการ เช่น นำข้อมูลภาพที่มีขนาดใหญ่ไปฝากไว้ที่อื่นแทนที่จะฝากไว้ในโฮสติ้งหรือการจัดการแคชของเว็บเบราว์เซอร์ เมื่อมีการเข้าเว็บครั้งถัดไป ก็ไปใช้ไฟล์เดิมที่เว็บเบราว์เซอร์เคยโหลดไว้ เป็นต้น ผู้ให้บริการโฮสติ้งฟรี เช่น Cloudflare Pages, Vercel และ Appwrite เป็นต้น โดยขั้นตอนการนำเว็บไซต์ไปเผยแพร่ (Deploy) จะแนะนำเฉพาะของ Cloudflare Pages

1) เตรียมไฟล์เว็บไซต์ HTML, JavaScript และ CSS ที่ต้องการเผยแพร่ อาจจะมีไฟล์ manifest.json และ sw.js สำหรับการเผยแพร่เว็บไซต์แบบ Progressive Web App (PWA) โดย PWA คือเว็บที่ทำให้มีการทำงานใกล้เคียงกับแอปพลิเคชันที่ถูกติดตั้งบนโทรศัพท์มือถือ ทำงานโดยใช้ เอนจินเบราว์เซอร์ (Browser Engine) ของแอปพลิเคชันเว็บเบราว์เซอร์ที่รองรับการทำงาน PWA เช่น Google Chrome, Microsoft Edge, Safari เป็นต้น ทำให้มีการทำงานที่ลื่นไหล เต็มหน้าจอ และติดตั้งง่าย โดยให้บีบอัดไฟล์ทั้งหมดลงในไฟล์ .zip เพื่อเตรียมนำไปเผยแพร่ต่อไป ดังภาพที่ 3.11

Name	Date modified	Type	Size
css	7/10/2567 13:16	File folder	
image	7/10/2567 13:16	File folder	
js	7/10/2567 13:16	File folder	
404	29/9/2567 21:47	Chrome HTML Do...	2 KB
climate	5/10/2567 20:29	Chrome HTML Do...	11 KB
climate-m	5/10/2567 20:28	Chrome HTML Do...	14 KB
googleab72a43c0caa992c			
index	5/10/2567 22:46	Chrome HTML Do...	5 KB
manifest	20/9/2567 11:10	JSON Source File	1 KB
navbar	1/10/2567 22:07	Chrome HTML Do...	12 KB
other	5/10/2567 23:24	Chrome HTML Do...	43 KB
semet	5/10/2567 23:24	Chrome HTML Do...	40 KB
sw.js	6/10/2567 17:17	JSFile	2 KB

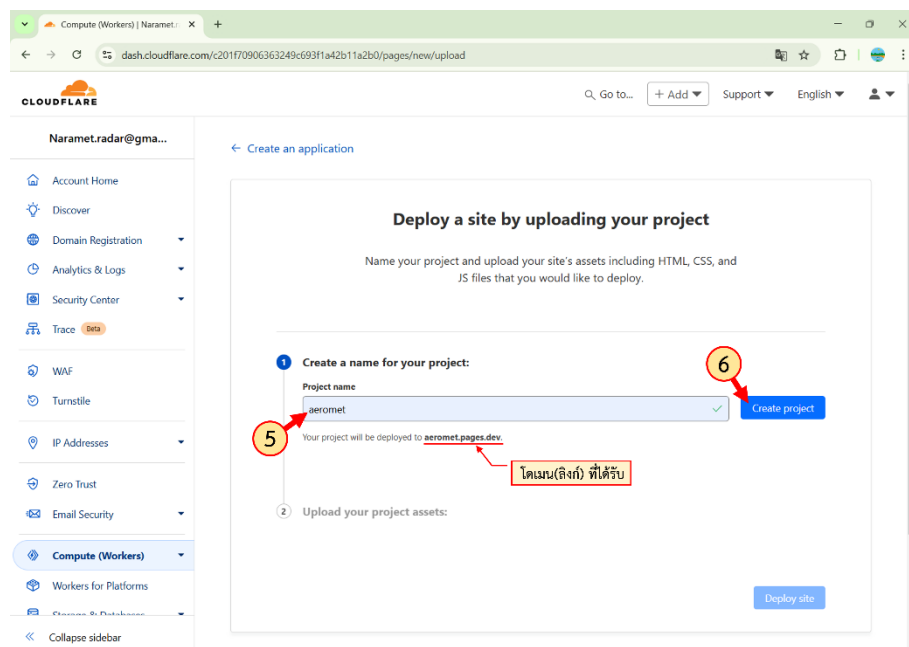
ภาพที่ 3.11 แสดงไฟล์ HTML, JavaScript และ CSS ที่ต้องการเผยแพร่

2) ไปยังเว็บไซต์ <https://dash.cloudflare.com> เมื่อทำการสมัครสมาชิกแล้ว ให้คลิกที่ “Compute (Workers)” จากนั้นคลิกที่ “Workers & Pages” ต่อไปเป็นการอัปโหลดไฟล์เว็บไซต์ ทำได้ 2 วิธี คือ การอัปโหลดไฟล์ไปที่ Cloudflare โดยตรง หรืออัปโหลดผ่าน GitHub สำหรับการอัปโหลดไฟล์โดยตรงนั้น ให้คลิกที่ “Upload assets” เพื่ออัปโหลดไฟล์ เหมาะสำหรับเว็บไซต์ที่มีการแก้ไขไม่บ่อย ส่วนการอัปโหลดผ่าน GitHub นั้น ให้คลิก “Connect to Git” สำหรับดึง Repository จาก GitHub เหมาะสำหรับเว็บไซต์ที่มีการแก้ไขบ่อย โดยจะทำการอัปโหลดเว็บไซต์ที่ถูกแก้ไขแล้วไปยัง GitHub และเผยแพร่โดยอัตโนมัติ ผ่านโปรแกรม GitHub Desktop พร้อมทั้งจัดเก็บเวอร์ชันก่อนๆไว้ ซึ่งในที่นี้จะกล่าวเฉพาะการ “Upload assets” ดังภาพที่ 3.12

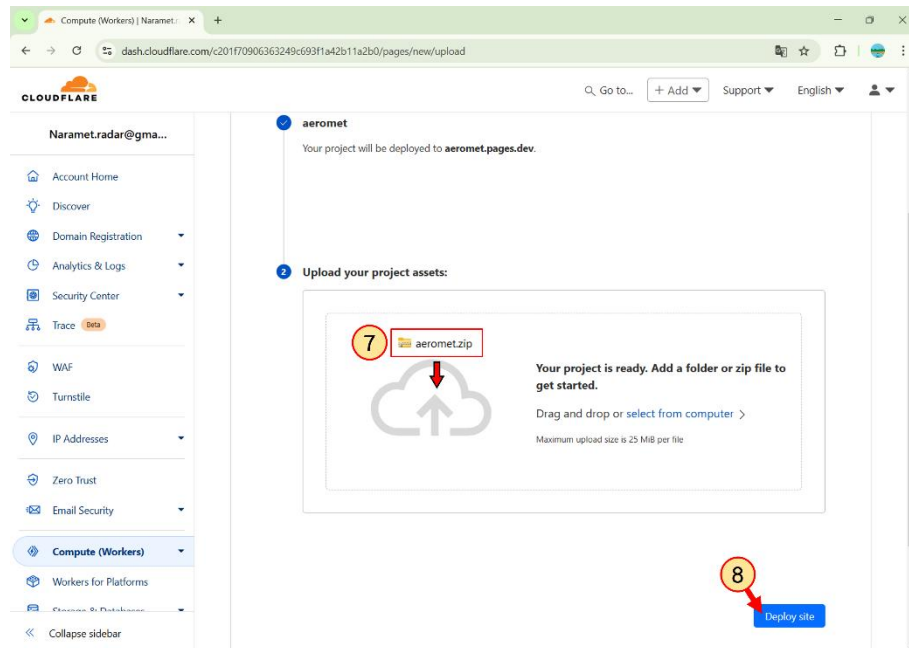


ภาพที่ 3.12 แสดงเว็บไซต์ <https://dash.cloudflare.com>

3) ตั้งชื่อเว็บไซต์ที่ต้องการ โดยจะได้รับชื่อโดเมนเป็น <ชื่อที่ตั้ง>.pages.dev แล้วคลิกปุ่ม “Create project” ดังภาพที่ 3.13 แล้วให้ทำการอัปโหลดไฟล์ .zip ในขั้นตอนที่ 1) ลงในช่อง “Upload your project assets” ดังภาพที่ 3.14 จากนั้นคลิก “Deploy site”

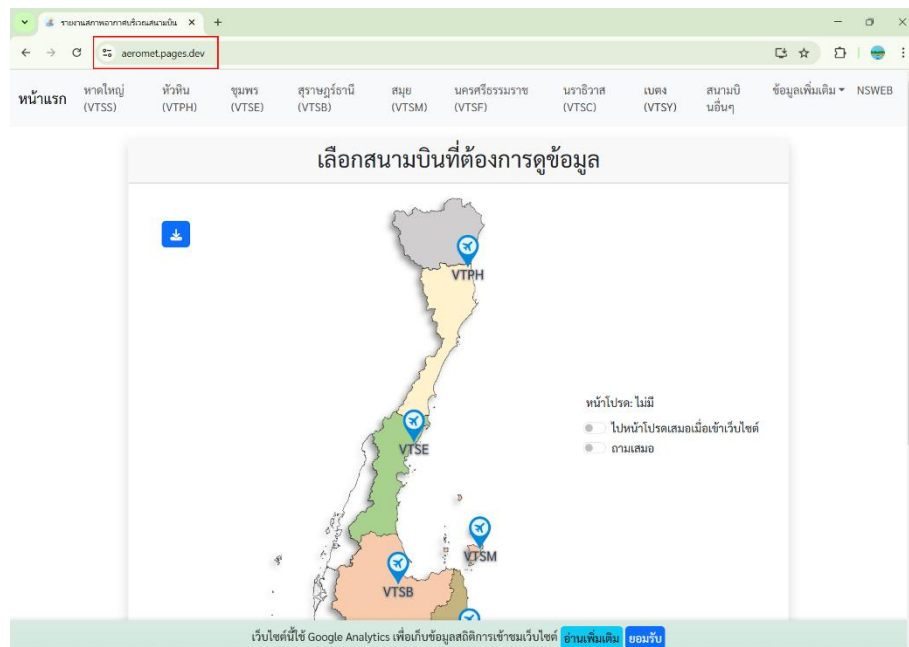


ภาพที่ 3.13 แสดงขั้นตอนการตั้งชื่อเว็บไซต์



ภาพที่ 3.14 แสดงขั้นตอนการอัปโหลดและเผยแพร่เว็บไซต์

4) ทดสอบเข้าเว็บไซต์ จากโดเมน (ลิงก์) ที่ได้รับในขั้นตอนที่ 3 จะได้เว็บไซต์ที่ออกแบบและเผยแพร่ ดังภาพที่ 3.15



ภาพที่ 3.15 แสดงเว็บไซต์ที่เผยแพร่

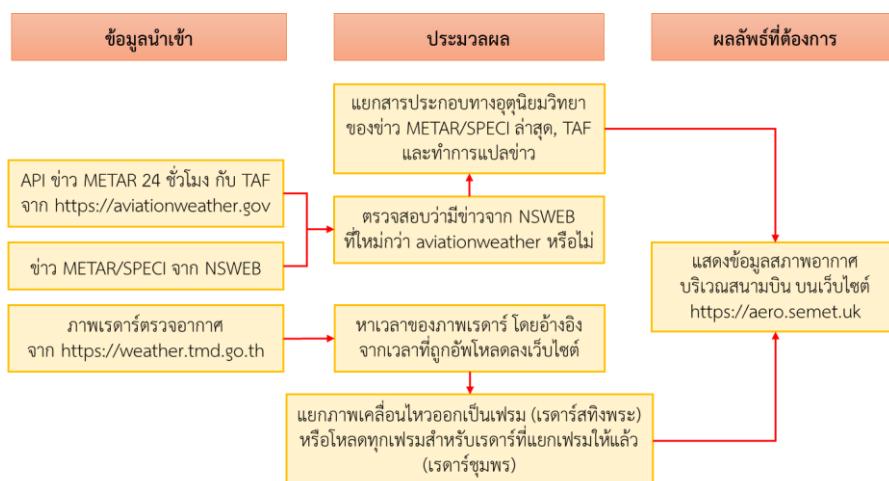
บทที่ 4

การพัฒนาเว็บไซต์สำหรับรายงานสภาพอากาศบริเวณสนามบิน

4.1 การพัฒนาเว็บไซต์รายงานสภาพอากาศบริเวณสนามบิน

ในการพัฒนาเว็บเพื่อรายงานสภาพอากาศบริเวณสนามบินนั้น มีเป้าหมายเพื่อแปลข่าวสภาพอากาศบริเวณสนามบิน (ข่าว METAR/SPECI) เพื่อให้บุคคลทั่วไป ที่ไม่ได้ปฏิบัติงานด้านอากาศการบิน รู้และเข้าใจสภาพอากาศบริเวณสนามบิน และยังสามารถใช้ภาพเรดาร์เพื่อหาระยะทางของฝนได้ นอกจากนี้ยังมีข้อมูลที่เป็นประโยชน์ต่อผู้ที่ปฏิบัติงานด้านอากาศการบิน ทั้งข่าวพยากรณ์แนวโน้มบริเวณสนามบิน (TAF) ข่าว METAR ย้อนหลัง 24 ชั่วโมง และข้อมูลภูมิอากาศของสนามบินในพื้นที่ภาคใต้ฝั่งตะวันออก

โดยมีข้อมูลนำเข้า การประมวลผล และผลลัพธ์ที่ต้องการ ดังภาพที่ 4.1

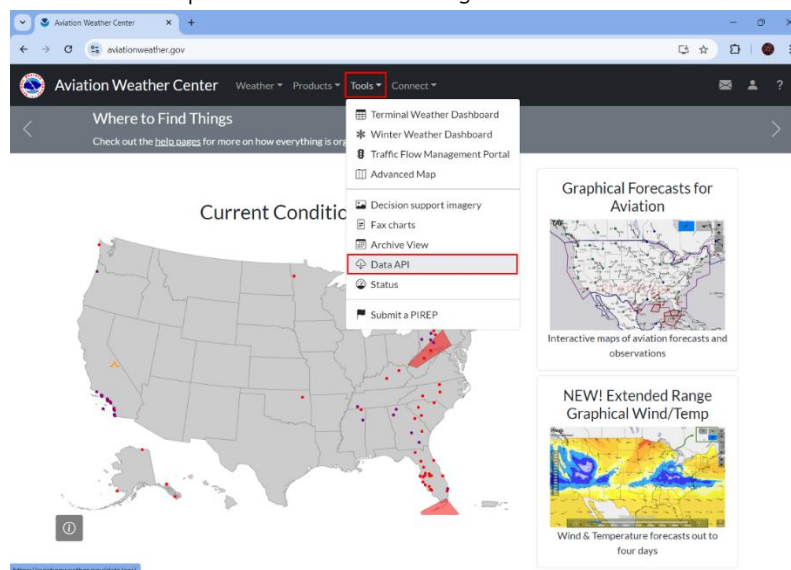


ภาพที่ 4.1 แสดงแผนผังข้อมูลนำเข้า การประมวลผล และผลลัพธ์ที่ต้องการในการพัฒนาเว็บเพื่อรายงานสภาพอากาศบริเวณสนามบิน

4.2 ข้อมูลนำเข้า

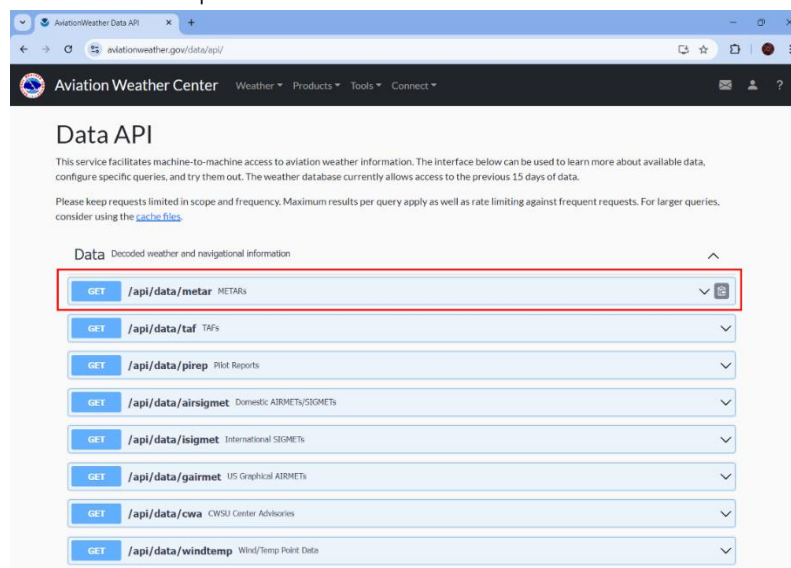
4.2.1 API ข่าว METAR กับ TAF จาก <https://aviationweather.gov> ซึ่งให้บริการ API สำหรับขอข้อมูลข่าว METAR TAF และอื่นๆ โดย Application Programming Interface (API) คือ ชุดคำสั่งหรือโปรโตคอลที่ช่วยให้ซอฟต์แวร์หรือแอปพลิเคชันต่างๆ สื่อสารกันได้ เช่น การใช้ API ของโซเชียลมีเดียเพื่อดึงข้อมูลโพสต์ หรือการใช้ API ของบริการการชำระเงินในการทำธุรกรรมออนไลน์ ระหว่างแอปพลิเคชันธนาคารกับเซิร์ฟเวอร์ของธนาคาร เป็นต้น โดยมีขั้นตอนการเรียก API ดังนี้

1) เข้าไปที่ <https://aviationweather.gov/> คลิกที่ Tools -> Data API



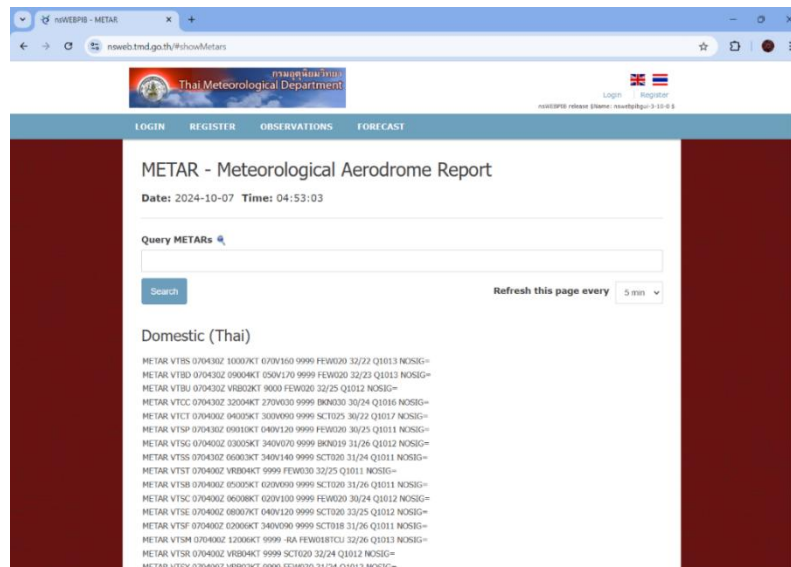
ภาพที่ 4.2 แสดงเว็บไซต์ <https://aviationweather.gov>

2) คลิกเลือก `/api/data/metar`



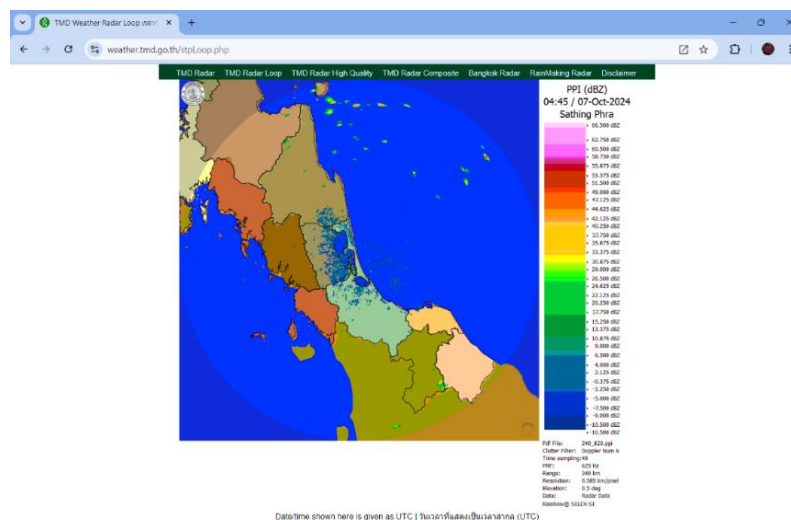
ภาพที่ 4.3 แสดง API ที่ aviationweather.gov ให้บริการ

4.2.2 ข่าว METAR/SPECI จาก <https://nsweb.tmd.go.th/#showMetars> ของกรมอุตุนิยมวิทยา



ภาพที่ 4.5 แสดงเว็บไซต์ <https://nsweb.tmd.go.th/#showMetars>

4.2.3 ภาพเรดาร์ตรวจอากาศจาก <https://weather.tmd.go.th> ของกรมอุตุนิยมวิทยา



ภาพที่ 4.6 แสดงเว็บไซต์ <https://weather.tmd.go.th>

4.3 การประมวลผลข้อมูลและการสร้างหน้าเว็บไซต์

ทำการประมวลผลข้อมูล และสร้างหน้าเว็บไซต์เพื่อแสดงข้อมูล โดยใช้โปรแกรม Visual Studio Code หรือ VS Code คือโปรแกรมแก้ไขโค้ด (Code Editor) แสดงดังภาพที่ 4.7 ประกอบกับการใช้ Node.js ซึ่งเป็น JavaScript runtime environment เป็นสิ่งที่จำเป็นในการพัฒนาเว็บไซต์ที่ใช้ JavaScript framework และใช้พัฒนา/เป็นระบบหลังบ้านได้ด้วย ซึ่งการพัฒนาเว็บไซต์ในครั้งนี้ มีความซับซ้อนระดับหนึ่ง ทำให้การพัฒนาโดยใช้เพียง HTML, CSS และ JavaScript นั้น ไม่เพียงพอ จึงต้องใช้ JavaScript framework ในการพัฒนา โดย JavaScript framework, ไลบรารี JavaScript และ CSS ที่สำคัญ มีดังนี้

- 1) Astro สำหรับสร้างหน้าเว็บไซต์ที่ฝั่งเซิร์ฟเวอร์ (Server-Side Rendering: SSR) เพื่อการทำ SEO ที่ดี ทำให้ Google Search เข้าถึงเนื้อหาของเว็บไซต์ได้ดียิ่งขึ้น อีกทั้งยังช่วยลดภาระการสร้างหน้าเว็บที่ฝั่งเบราว์เซอร์ และซ่อนโค้ดบางส่วนได้
- 2) React เป็น UI Framework สำหรับสร้างหน้าเว็บไซต์ ที่ฝั่งไคลน์เอ็นเป็นหลัก (Client-Side Rendering: CSR) สามารถสร้างที่ฝั่งเซิร์ฟเวอร์ได้เช่นกัน แต่ต้องมี SSR Framework ช่วย เน้นการทำงานแบบ component-based คือการสร้างชิ้นส่วนย่อยที่สามารถนำกลับมาใช้ซ้ำได้เรื่อย ๆ และใช้ Virtual DOM เพื่ออัปเดตหน้าจออย่างมีประสิทธิภาพ
- 3) Bootstrap สำหรับจัดหน้าเว็บไซต์ให้เป็นระเบียบ และปรับขนาดของแต่ละส่วนประกอบในหน้าเว็บไซต์ตามอุปกรณ์ (Responsive Web Design)
- 4) jQuery คือไลบรารี JavaScript ที่ทำให้การเขียนโค้ด JavaScript ง่ายขึ้น โดยช่วยให้สามารถจัดการกับองค์ประกอบในหน้าเว็บ ทำงานกับ DOM, จัดการเหตุการณ์ (events), และทำเอฟเฟคต่างๆ ได้ง่ายขึ้น
- 5) metar-taf-parser สำหรับแยกส่วนข่าว METAR ให้เป็นส่วนๆ (ไม่ได้แปลข่าว)
- 6) gifuct-js สำหรับแยกเฟรมไฟล์ภาพเคลื่อนไหว Gif ของเรดาร์สทิงพระ
- 7) font-awesome สำหรับแสดงไอคอน สัญลักษณ์ต่างๆ

มีขั้นตอนการดำเนินการ ดังนี้

- 1) เลือกสนามบินที่ต้องการจะทำการแปลข่าว โดยใช้รหัส ICAO แล้วทำการดึงข้อมูลข่าว METAR ปัจจุบัน และย้อนหลัง 24 ชั่วโมง พร้อมทั้งข่าว TAF จากเว็บไซต์ aviationweather.gov และดึงข่าว METAR/SPECI จากเว็บไซต์ nsweb.tmd.go.th แล้วทำการเปรียบเทียบเวลาข่าว ทั้ง 2 เว็บไซต์ เลือกข่าวที่รายงานล่าสุดมาทำการแปล

2) ใช้ไลบรารี `metar-taf-parser` แยกส่วนข่าว METAR/SPECI และ TAF ให้เป็นแต่ละสารประกอบอุตุนิยมวิทยา แล้วทำการแปลเป็นภาษาไทย (หรืออังกฤษ) ในแต่ละส่วน พร้อมกับมีภาพกราฟิกประกอบ โดยข้อมูลประกอบด้วย ประเภทข่าว ชื่อสนามบิน เวลาที่ส่งข่าว ทิศทางและความเร็วลม ทิศวิสัย สภาพอากาศปัจจุบัน จำนวนและความสูงของฐานเมฆ อุณหภูมิ อุณหภูมิจุดน้ำค้าง ความกดอากาศ QNH ระยะ RVR และข่าวสารเพิ่มเติม

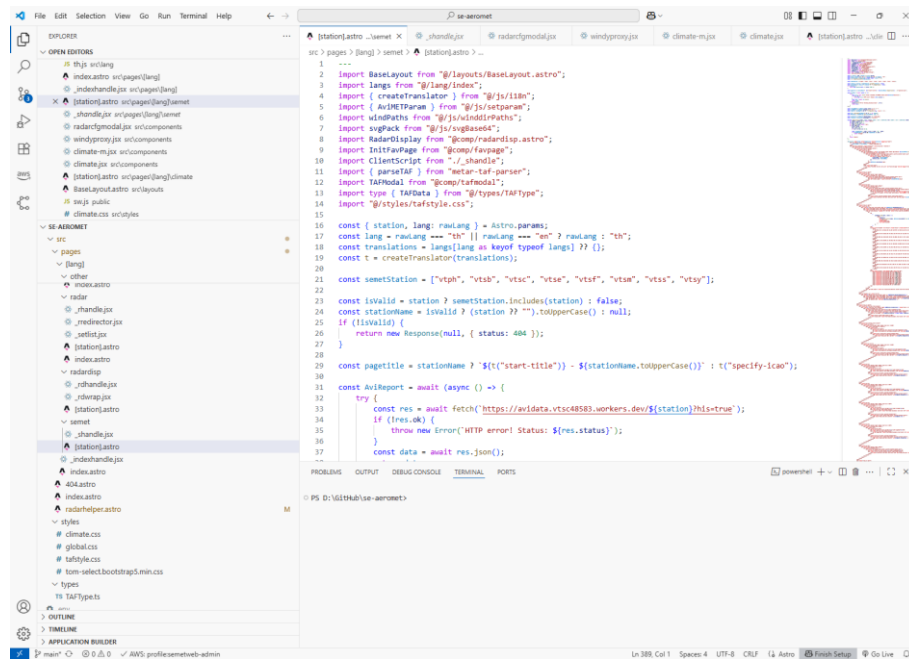
3) ในส่วนของการแสดงผล จะออกแบบโดยใช้ภาพกราฟิกแบบเวกเตอร์ (Scalable Vector Graphics: SVG) คือ รูปแบบไฟล์กราฟิกที่ใช้ XML ในการกำหนดโครงสร้างภาพสามารถขยายหรือย่อขนาดได้โดยไม่สูญเสียความคมชัด ทั้งยังสามารถใช้งานได้โดยตรงใน HTML และ CSS และใช้ JavaScript หรือ CSS ควบคุมการเปลี่ยนแปลง เคลื่อนไหวได้

4) ใช้ไลบรารี `Regex` (Regular Expression) คือ รูปแบบของข้อความที่ใช้สำหรับค้นหาและจัดการสตริง (String) โดยใช้กฎหรือรูปแบบเฉพาะ ซึ่งช่วยให้สามารถตรวจสอบ ตรวจสอบจับและแทนที่ข้อความได้อย่างมีประสิทธิภาพ ใช้ในการค้นหาและแยกบางส่วนของข่าว ที่ไลบรารี `metar-taf-parser` ไม่รองรับ

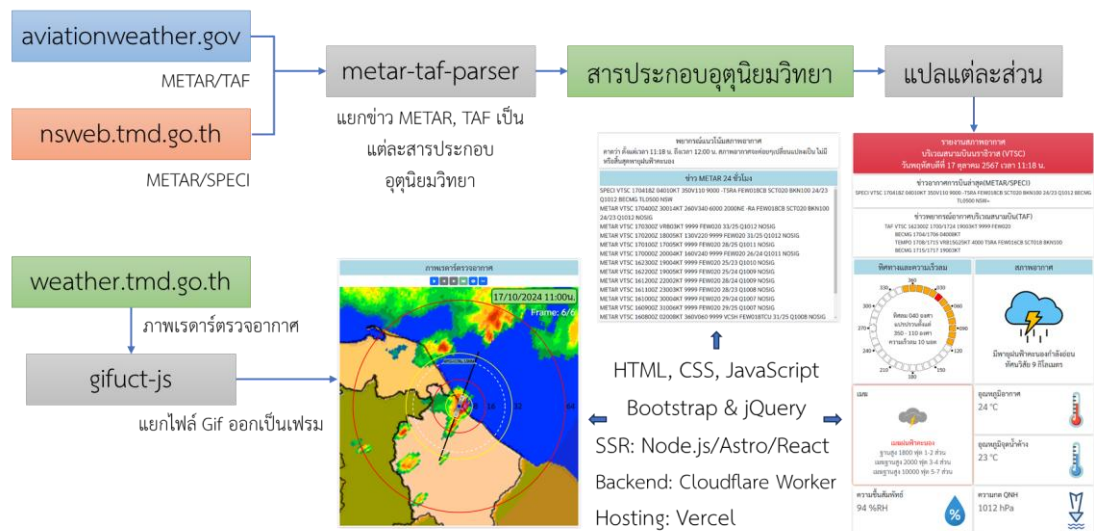
5) ทำการดึงภาพเรดาร์จาก `weather.tmd.go.th` มาแสดงเฉพาะบริเวณสนามบิน โดยได้ออกแบบการควบคุมการเคลื่อนไหวของแต่ละเฟรม สามารถย่อ-ขยาย และสามารถวัดระยะได้ โดยเทียบอัตราส่วนระหว่างระยะทางจริง (ระยะตรวจอากาศรัศมี 240 กิโลเมตร) กับขนาดพิกเซล (Pixel) ของภาพเรดาร์นั้นๆ ถ้าภาพเรดาร์เป็นแบบเคลื่อนไหว (Loop) ที่อยู่ในรูปแบบไฟล์อนิเมชัน Gif จะใช้ไลบรารี `gifuct-js` ในการแยกส่วนภาพเรดาร์แต่ละเฟรม

6) ออกแบบและพัฒนาเว็บไซต์ โดยใช้ `Astro` และ `React` สร้างหน้าเว็บ (Render) ให้เป็นภาษา HTML ร่วมกับการใช้ CSS และ JavaScript ใช้ไลบรารี `Bootstrap` และ `jQuery` ช่วยในการออกแบบเว็บไซต์ให้สวยงาม และสามารถตอบโต้กับผู้ใช้งานได้ โดยจะแบ่งเว็บไซต์ออกเป็น 3 ส่วน คือ ส่วนแสดงการแปลข่าวอากาศการบิน (METAR/SPECI) ในรูปแบบกราฟิกที่เข้าใจได้ง่าย และสำหรับสภาพอากาศปัจจุบัน จะแสดงภาพกราฟิกที่มีความหมายตามตารางที่ 4.1 พร้อมทั้งแปลข่าวพยากรณ์อากาศการบิน (TAF) ประกอบกับเส้นเวลากำกับเหตุการณ์ที่คาดว่าจะเกิดขึ้น ถัดไปคือ ส่วนแสดงภาพเรดาร์ตรวจอากาศบริเวณสนามบินนราธิวาส พร้อมกับแสดงวงกลมรัศมี 8 16 32 และ 64 กิโลเมตร แนวทางวิ่ง 20 และระยะเขตประชิดสนามบิน ซ้อนทับกับภาพเรดาร์ สามารถควบคุมการเคลื่อนไหวของภาพได้ และสามารถวัดระยะทางระหว่างจุดสองจุดบนภาพเรดาร์ได้ และสุดท้ายเป็นส่วนแสดงข่าว METAR ย้อนหลัง 24 ชั่วโมง

สามารถสรุปเครื่องมือ และขั้นตอนการดำเนินการได้ ดังภาพที่ 4.8 ซึ่งหลังจากที่ออกแบบและพัฒนาเว็บไซต์แล้ว จะทำการเผยแพร่เว็บไซต์โดยใช้ `Vercel` เป็นเว็บโฮสติ้ง และใช้ `Cloudflare Worker` เป็นระบบหลังบ้าน ซึ่งให้บริการฟรีแบบมีข้อจำกัด (สามารถใช้ `Vercel` เป็นระบบหลังบ้านได้เช่นกัน แต่ให้ `Cloudflare Worker` ช่วยแบ่งเบาภาระ ไม่ให้เกินโควตาแผนฟรี)

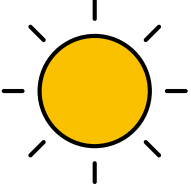
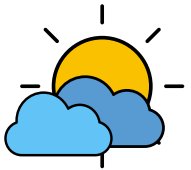


ภาพที่ 4.7 แสดงโปรแกรม Visual Studio Code



ภาพที่ 4.8 แผนภาพแสดงเครื่องมือ และขั้นตอนการดำเนินการทำเว็บไซต์

ตารางที่ 4.1 กราฟิกสภาพอากาศปัจจุบัน (Present Weather)

			
ท้องฟ้าแจ่มใส (เช้า)	ท้องฟ้าแจ่มใส (ค่ำ)	เมฆเล็กน้อย (เช้า)	เมฆเล็กน้อย (ค่ำ)
			
เมฆบางส่วน (เช้า)	เมฆบางส่วน (ค่ำ)	เมฆส่วนมาก	เมฆเต็มท้องฟ้า
			
ฝนกำลังอ่อน	ฝนกำลังปานกลาง	ฝนกำลังแรง	พายุฟ้าคะนอง
			
พายุฝนฟ้าคะนอง กำลังอ่อน	พายุฝนฟ้าคะนอง กำลังปานกลาง	พายุฝนฟ้าคะนอง กำลังแรง	ฝนชุกบริเวณ ใกล้เคียง
			
ฟ้าหลัวขึ้น	หมอก	ฟ้าหลัวแห้ง	ควัน

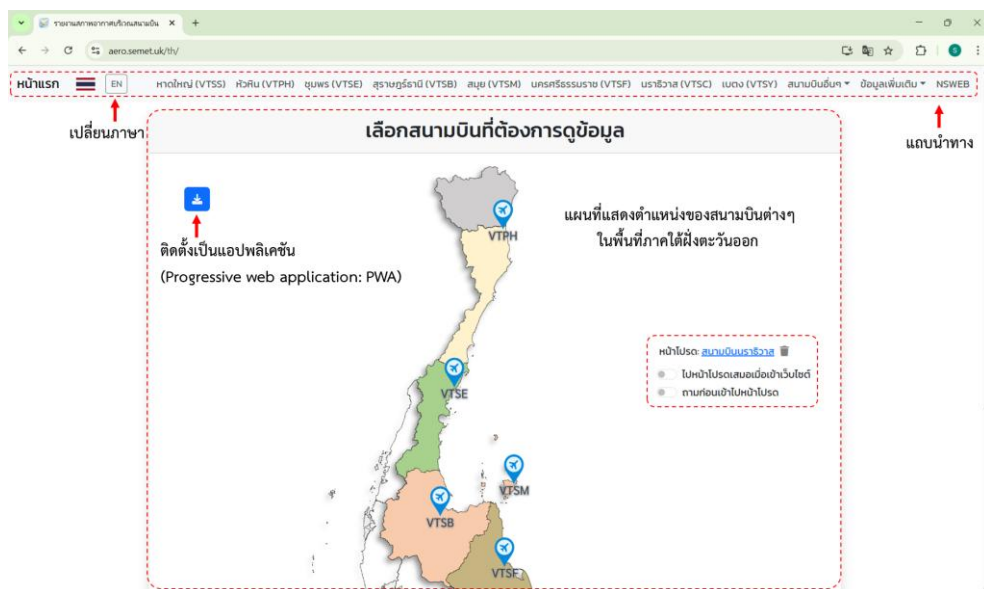
ที่มา: <https://freedesignresources.net/34-weather-icon-pack>

บทที่ 5

การใช้งานเว็บไซต์

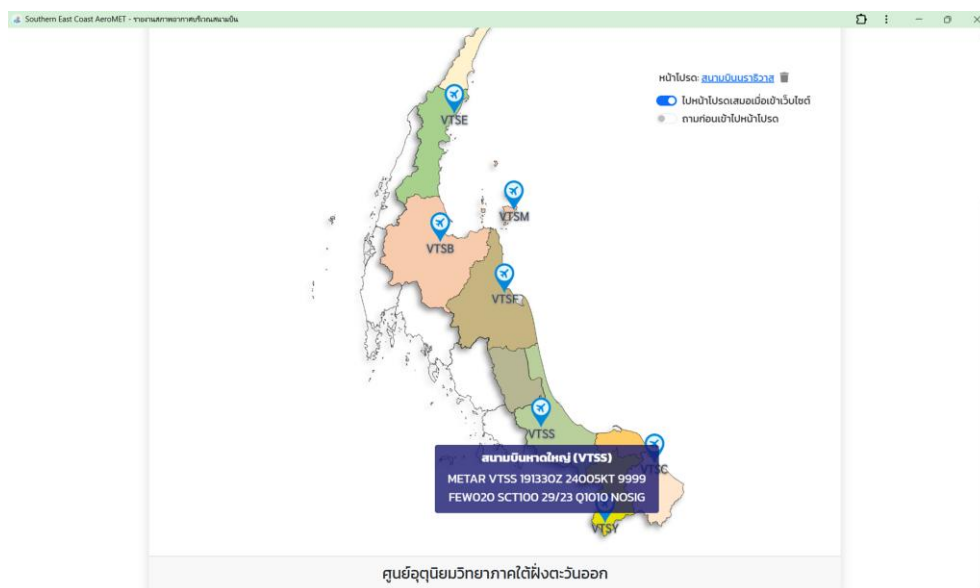
5.1 สนามบินในพื้นที่ภาคใต้ฝั่งตะวันออก

เข้าไปที่เว็บไซต์ <https://aero.semet.uk> ดังภาพที่ 5.1 ประกอบไปด้วยแถบนำทางสำหรับเลือกสนามบินที่ต้องการดูข้อมูลสภาพอากาศ หรือสามารถคลิกเลือกสนามบินบนแผนที่ก็ได้ ซึ่งเว็บไซต์สามารถแสดงผลได้ 2 ภาษา คือภาษาไทย และภาษาอังกฤษ โดยการคลิกที่ปุ่มเปลี่ยนภาษา (ปุ่ม TH/EN) และยังสามารถติดตั้งเป็นแอปพลิเคชันแบบ Progressive web application (PWA) ได้ และยังสามารถกำหนดหน้าโปรด เพื่อให้เข้าถึงสนามบินที่ต้องการดูข้อมูลได้ง่ายขึ้น



ภาพที่ 5.1 แสดงเว็บไซต์ <https://aero.semet.uk>

เมื่อนำเมาส์ไปไว้เหนือตำแหน่งสนามบินบนแผนที่ จะปรากฏข่าวรายงานสภาพอากาศ (METAR/ SPECI) ของสนามบินนั้นๆ ดังภาพที่ 5.2 โดยข่าว METAR จะแสดงเป็นสีน้ำเงิน ข่าว SPECI จะแสดงเป็นสีแดง และหากไม่มีการรายงานข่าว จะแสดงเป็นสีดำ โดยทางด้านขวาของแผนที่ จะแสดง “หน้าโปรด” ที่บันทึกไว้ และปุ่ม “ไปหน้าโปรดเสมอเมื่อเข้าเว็บไซต์” หมายถึงจะไปยังหน้าโปรดทันทีที่เข้าเว็บไซต์หรือแอปพลิเคชัน เมื่อเปิดครั้งแรก ส่วน “ถามก่อนเข้าไปหน้าโปรด” หากเปิดไว้ จะมีข้อความถามเมื่อเข้ามายังหน้าหลักนี้ หรือเมื่อทำการรีเฟรชหน้าจอ



ภาพที่ 5.2 แสดงข่าวรายงานสภาพอากาศ (METAR/SPECI) ที่ปรากฏขึ้นมา

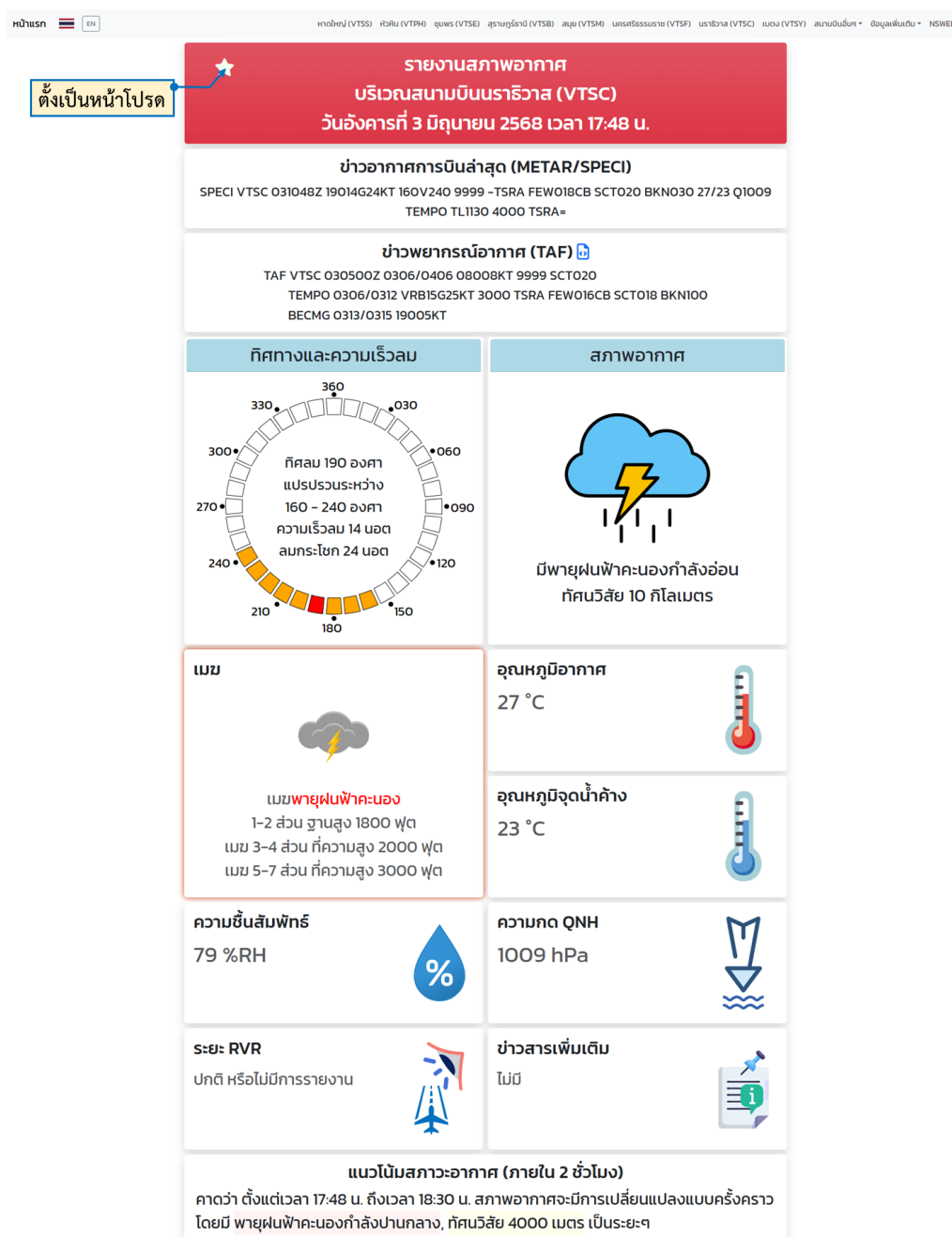
เมื่อคลิกเลือกสนามบินที่ต้องการดูข้อมูลสภาพอากาศแล้ว จะไปยังหน้าเว็บไซต์ดังภาพที่ 5.4 ถึงภาพที่ 5.6 มีส่วนประกอบ และรายละเอียดดังนี้

ส่วนที่ 1: แสดงการแปลข่าวอากาศการบิน (METAR/SPECI) ให้เข้าใจได้ง่าย ดังภาพที่ 5.4 ประกอบด้วย

- 1) ชื่อสนามบินพร้อมรหัส ICAO และเวลาที่รายงานข่าว METAR/SPECI โดยจะมีพื้นหลังสีเขียวเมื่อรายงานข่าว METAR และสีแดงเมื่อรายงานข่าว SPECI ดังภาพที่ 5.3 และมีไอคอนรูปดาว เพื่อกำหนดหรือลบหน้าโปรต
- 2) รหัสข่าวอากาศการบินล่าสุด (METAR/SPECI)
- 3) รหัสข่าวพยากรณ์อากาศบริเวณสนามบิน (TAF) และปุ่มแปลรหัสข่าว
- 5) กราฟิกระบบแสดงผล ลมเฉลี่ย 10 นาที
- 5) สภาพอากาศปัจจุบัน และทัศนวิสัย ถ้าหากทัศนวิสัยต่ำกว่า 5000 เมตร จะแสดงกรอบเป็นสีเหลือง แล้วถ้าหากต่ำกว่า 3000 เมตร จะเป็นสีแดง
- 6) จำนวน ความสูงของเมฆ และชนิดเมฆ (ถ้ามี CB/TCU) ถ้าเป็น CB จะแสดงกรอบเป็นสีแดง แล้วถ้าเป็น TCU จะเป็นสีเหลือง
- 7) อุณหภูมิ
- 8) อุณหภูมิจุดน้ำค้าง
- 9) ความชื้นสัมพัทธ์ ที่คำนวณจากอุณหภูมิอากาศและอุณหภูมิจุดน้ำค้าง
- 10) ความกดอากาศ QNH
- 11) ระยะทางวิ่ง (RVR) พร้อมกับแนวโน้มน
- 12) ข่าวสารเพิ่มเติม (สภาพอากาศที่ผ่านมา กับ การรายงานลมเฉือน)
- 13) การแปลพยากรณ์แนวโน้มสภาวะอากาศ (Trend Forecast)

- ★ รายงานสภาพอากาศ
บริเวณสนามบินหาดใหญ่ (VTSS)
วันอังคารที่ 3 มิถุนายน 2568 เวลา 15:30 น.
- ★ รายงานสภาพอากาศ
บริเวณสนามบินหาดใหญ่ (VTSS)
วันอังคารที่ 3 มิถุนายน 2568 เวลา 15:04 น.

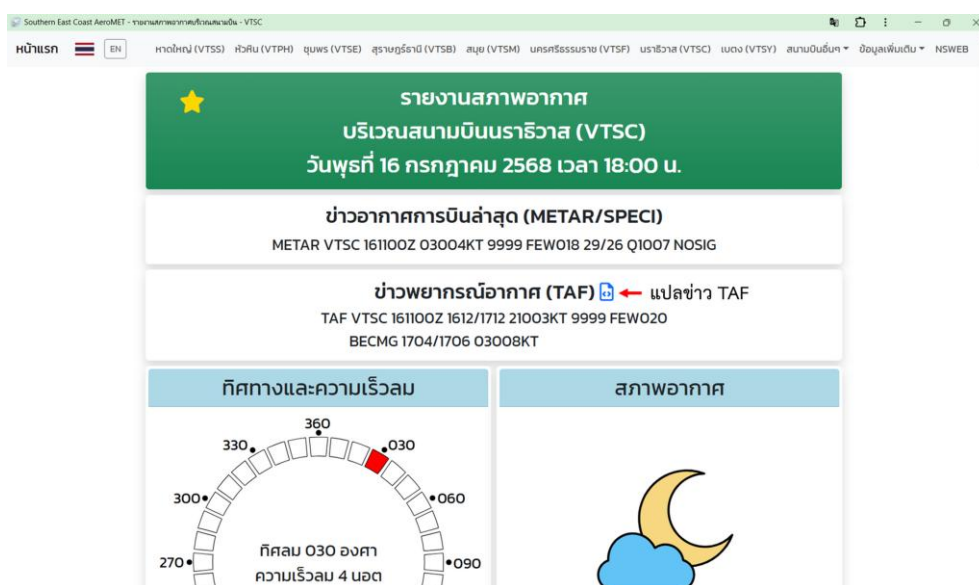
ภาพที่ 5.3 แสดงสี่พื้นหลังของการรายงานข่าว METAR (บน) และ SPECI (ล่าง)



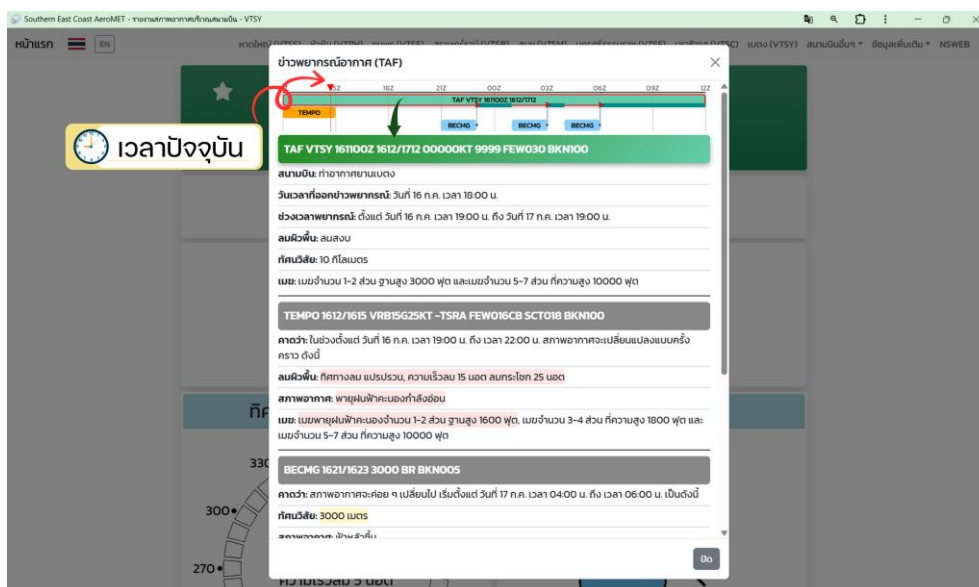
ภาพที่ 5.4 แสดงเว็บไซต์ส่วนแสดงการแปลข่าวอากาศการบิน (METAR/SPECI)

ส่วนที่ 2: การแปลข่าวพยากรณ์อากาศบริเวณสนามบิน (TAF) เมื่อคลิกปุ่มตามภาพที่ 5.5 จะแสดงส่วนของการแปลข่าว TAF ดังภาพที่ 5.6

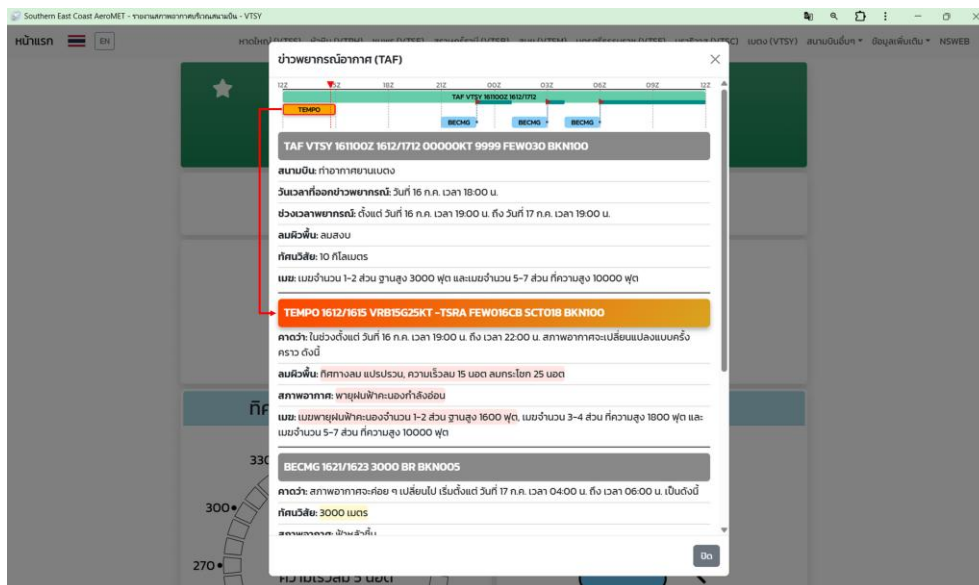
- 1) เส้นเวลาแสดงลำดับเหตุการณ์สภาพอากาศที่คาดว่าจะเกิดขึ้น ตามช่วงเวลา ที่ข่าว TAF ครอบคลุม (24/30 ชั่วโมง) รวมถึงคาดการณ์การเปลี่ยนแปลงที่ มีนัยสำคัญ (หากมี) ประกอบไปด้วย BECMG, FM, TEMPO, PROB_{C₂C₂}, PROB_{C₂C₂} TEMPO และมีเส้นระบุเวลาปัจจุบัน (เส้นสีแดง)
- 2) แสดงการแปลข่าว TAF ส่วนต่างๆ ถ้าหากเอาเมาส์วางไว้เหนือเส้นเวลาที่ ส่วนใด ก็จะมี (เปลี่ยนสี) การแปลข่าวของส่วนนั้นให้เด่นชัด ดังภาพที่ 5.7



ภาพที่ 5.5 แสดงปุ่มกด เพื่อแสดงการแปลข่าวพยากรณ์อากาศบริเวณสนามบิน (TAF)

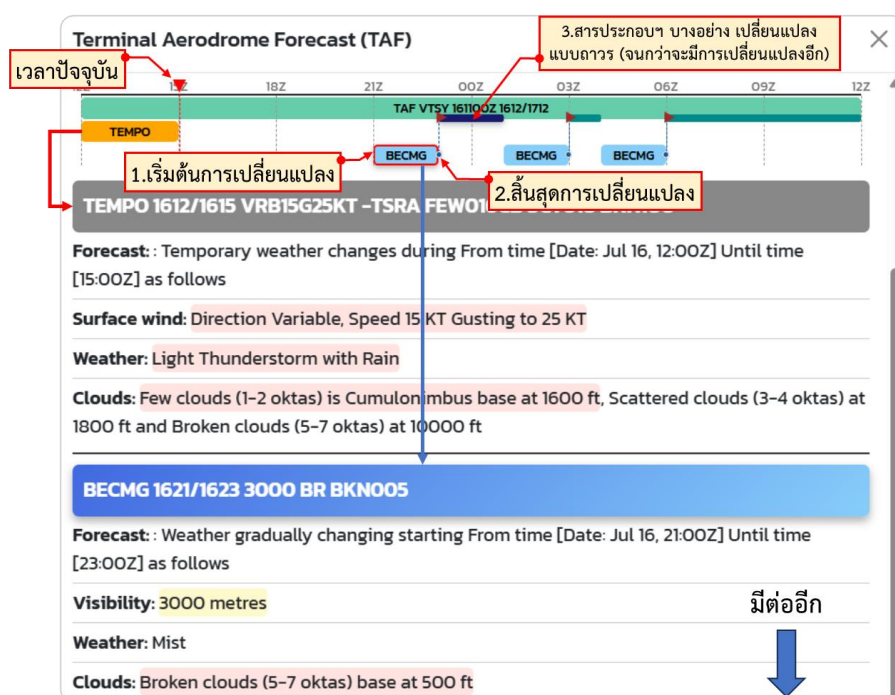


ภาพที่ 5.6 แสดงการแปลข่าวพยากรณ์อากาศบริเวณสนามบิน (TAF)



ภาพที่ 5.7 แสดงการเน้น (เปลี่ยนสี) แบบจับคู่ของเส้นเวลา และการแปลข่าว TAF

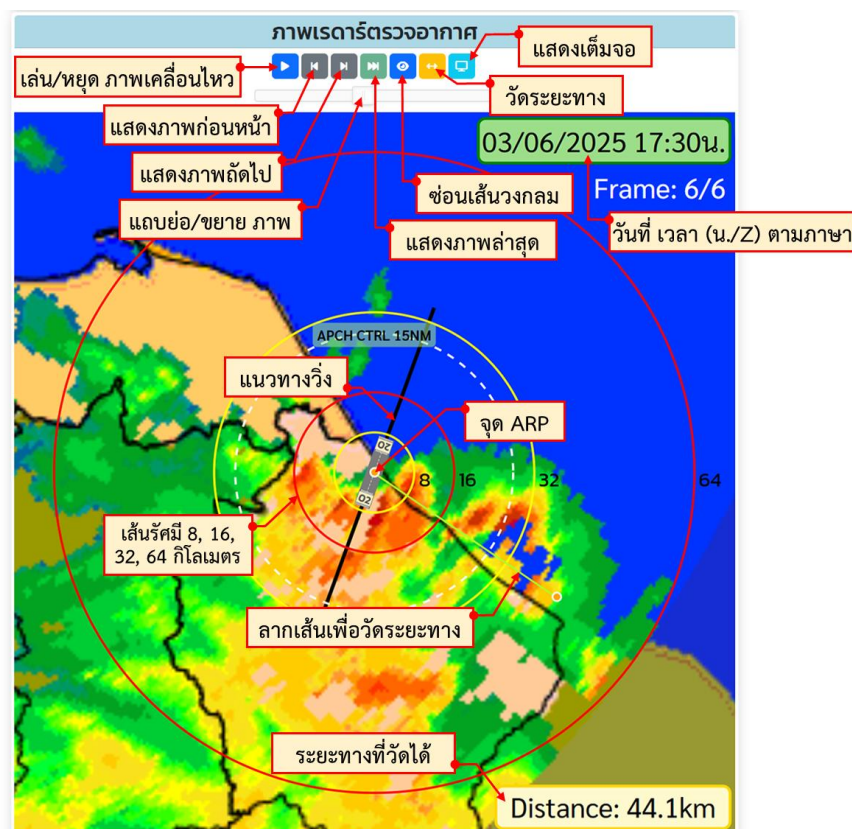
จากภาพที่ 5.7 จะเห็นว่า เมื่อเอาเมาส์วางไว้เหนือส่วน TEMPO ที่เส้นเวลาแล้ว ส่วน TEMPO พร้อมการแปลของส่วนนั้น จะเด่นชัดขึ้น แล้วถ้าหากคลิกที่เส้นเวลาที่ส่วนใด ก็จะไปแสดงการแปลข่าวของส่วนนั้น เช่น คลิกที่ส่วน BECMG ก็จะไปหาส่วนที่แปล BECMG นั้น และจากภาพที่ 5.8 ให้สังเกตที่ด้านหลังส่วน BECMG บนเส้นเวลา จะเห็นเส้นเล็กๆ ขนานไปพร้อมกับส่วนของ TAF บ่งบอกว่า เมื่อถึงจุดสิ้นสุดเปลี่ยนแปลงการเปลี่ยนแปลงแบบ BECMG แล้ว จะเป็นการเปลี่ยนแปลงแบบถาวร (เปลี่ยนแปลงเฉพาะสารประกอบทางอุตุนิยมวิทยาที่ระบุไว้ในส่วนของ BECMG นั้นๆ) ตามนิยามของการคาดการณ์การเปลี่ยนแปลงแบบ BECMG



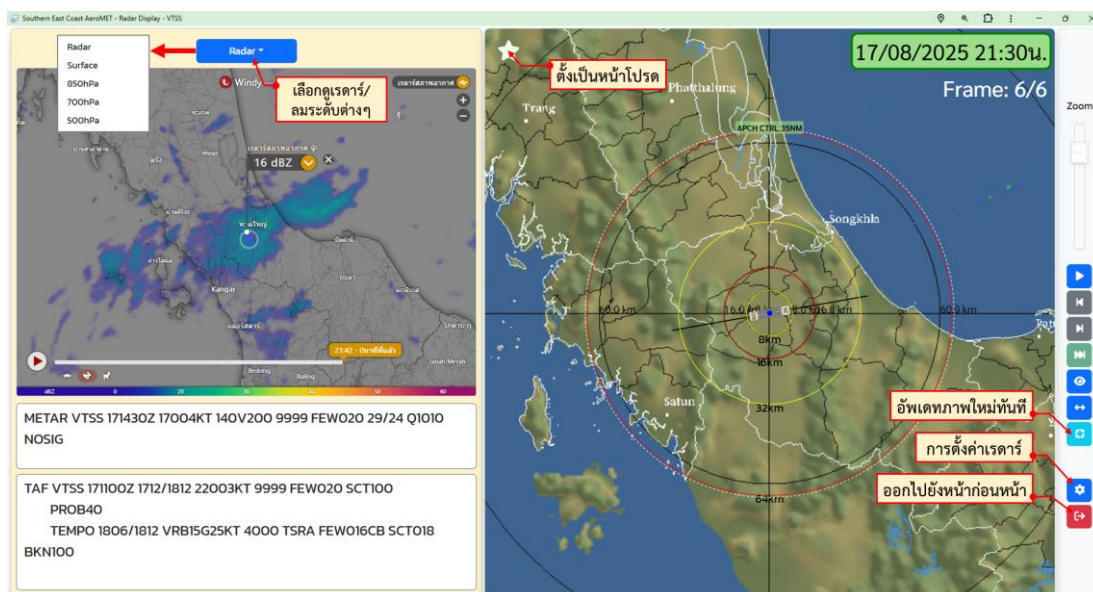
ภาพที่ 5.8 แสดงการคาดการณ์การเปลี่ยนแปลงแบบ BECMG

ส่วนที่ 3: แสดงภาพเรดาร์ตรวจอากาศบริเวณสนามบินราธิวาส ดังภาพที่ 5.9 ประกอบด้วย

- 1) เส้นแนวทางวิ่ง 20, ตำแหน่งที่ตั้งสนามบินราธิวาส, เส้นวงกลมรัศมี 8 16 32 64 กิโลเมตร และระยะเขตประชิดสนามบิน (Approach Control)
- 2) สามารถควบคุมการเคลื่อนไหวของภาพเรดาร์แบบเคลื่อนไหว (Loop) ได้ เพื่อวิเคราะห์การเคลื่อนตัวของกลุ่มเมฆฝน โดยการคลิกปุ่มหรือให้เล่นอัตโนมัติ
- 3) สามารถซ่อน/แสดง เส้นวงกลมต่างๆ เส้นแนวทางวิ่งได้
- 4) สามารถวัดระยะห่างระหว่าง 2 จุดได้ (หน่วยกิโลเมตร)
- 5) สามารถแสดงภาพเรดาร์แบบเต็มจอได้ สำหรับใช้ในการปฏิบัติงานของเจ้าหน้าที่อุตุนิยมวิทยา (อุปกรณ์ต้องมีจอกว้าง $\geq 992\text{px}$) ดังภาพที่ 5.10



ภาพที่ 5.9 แสดงเว็บไซต์ส่วนภาพเรดาร์ตรวจอากาศ บริเวณสนามบินราธิวาส



ภาพที่ 5.10 แสดงภาพเรดาร์ตรวจอากาศแบบเต็มจอ บริเวณสนามบินหาดใหญ่

จากภาพที่ 5.10 จะประกอบไปด้วยส่วนต่างๆ ซึ่งในหน้าเว็บไซต์นี้ จะมีการอัปเดตแต่ ละส่วนแบบอัตโนมัติ (1) ภาพเรดาร์ตรวจอากาศ จะอัปเดตตามเวลาที่เหมาะสม เช่น เรดาร์ของกรม อุตุนิยมวิทยา จะอัปเดตช้ากว่าเวลาจริงไปประมาณ 4 - 6.5 นาที จึงให้อัปเดตในเว็บไซต์ทุกๆ นาทีที่ 7 22 37 และ 52 ตามลำดับ (ทุกๆ 15 นาที) แล้วเวลาที่กำกับภาพเรดาร์ จะถูกคำนวณโดยอ้างอิง จากเวลาจริง ที่กรมอุตุนิยมวิทยาอัปโหลดภาพไปที่เซิร์ฟเวอร์ เพื่อความแม่นยำ และถ้าหากเป็นเรดาร์ ของกรมฝนหลวงและการบินเกษตร จะอัปเดตทุกๆ 6 นาที (ตั้งแต่นาทีที่ 7 เป็นต้นไป) ซึ่งกรมฝน หลวงฯ ให้บริการ API ภาพเรดาร์แบบ License: Open Data Common มีรูปแบบข้อมูล API ที่ระบุ เวลาของแต่ละภาพไว้อย่างชัดเจน (2) ข่าว METAR/SPECI และ TAF จะอัปเดตทุกๆ 2 นาที (ในส่วน นี้ ผู้พัฒนาเว็บไซต์ได้จัดการแคช/สำรองข้อมูลต่างๆ ไว้อย่างเป็นระบบ เพื่อไม่ให้เจ้าของข้อมูลได้รับ ผลกระทบ) (3) ภาพเรดาร์ และ ลมระดับชั้นต่างๆ จากเว็บไซต์ www.windy.com จะอัปเดตทุกๆ 5 นาที ส่วนปุ่มการตั้งค่าเรดาร์ จะอธิบายรายละเอียดในหัวข้อที่ 5.2

หมายเหตุ หน้าเว็บไซต์ อาจมีการเปลี่ยนแปลงในภายหลัง

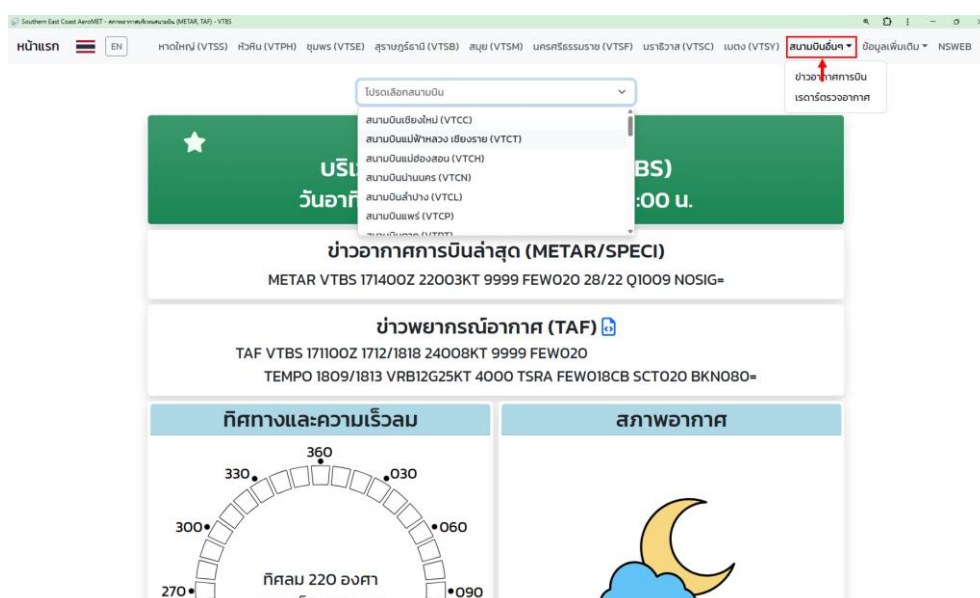
ส่วนที่ 4: แสดงข่าว METAR ย้อนหลัง 24 ชั่วโมง ดังภาพที่ 5.11

ข่าว METAR 24 ชั่วโมง
SPECI VTSC 031048Z 19014G24KT 160V240 9999 -TSRA FEW018CB SCT020 BKN030 27/23 Q1009 TEMPO TL1130 4000 TSRA
METAR VTSC 031000Z 35008KT 9999 FEW022 31/25 Q1008 TEMPO FM1030 4000 TSRA FEW018CB
METAR VTSC 030900Z 36007KT 9999 FEW022 SCT100 31/25 Q1008 NOSIG
METAR VTSC 030800Z 02008KT 350V050 9999 FEW022 31/25 Q1008 NOSIG
METAR VTSC 030700Z 04007KT 9999 FEW022 32/26 Q1008 NOSIG
METAR VTSC 030600Z 06006KT 030V090 9999 FEW022 32/25 Q1009 NOSIG
METAR VTSC 030500Z 06005KT 020V080 9999 FEW022 32/26 Q1010 NOSIG
METAR VTSC 030400Z VRB03KT 9999 FEW022 32/25 Q1010 NOSIG
METAR VTSC 030300Z 19003KT 100V260 9999 FEW022 31/25 Q1010 NOSIG
METAR VTSC 030200Z 15004KT 100V200 9999 FEW020 30/25 Q1010 NOSIG
METAR VTSC 030100Z 00000KT 9999 FEW018 28/26 Q1010 NOSIG
METAR VTSC 030000Z 00000KT 9999 FEW018 27/25 Q1010 NOSIG
METAR VTSC 022300Z 00000KT 9999 FEW018 27/25 Q1009 NOSIG

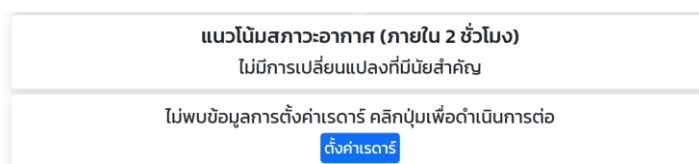
ภาพที่ 5.11 แสดงเว็บไซต์ส่วนแสดงข่าว METAR ย้อนหลัง 24 ชั่วโมง

5.2 สนามบินอื่นๆ

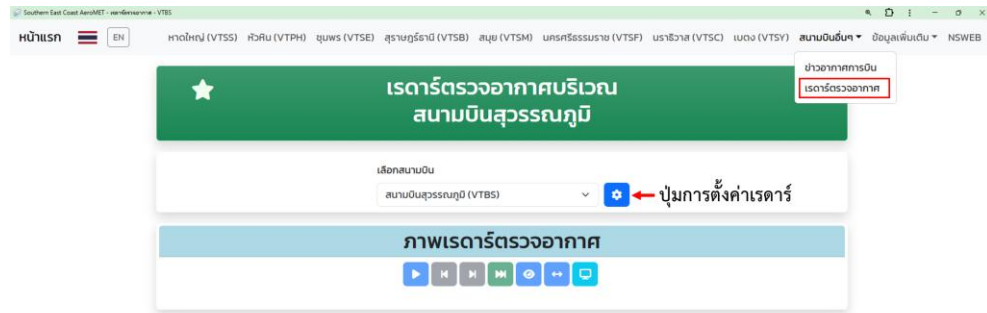
ให้คลิกที่แถบเมนู “สนามบินอื่นๆ” จากนั้นคลิกที่ “ข่าวอากาศการบิน” แล้วพิมพ์ค้นหาชื่อสนามบิน เว็บไซต์จะแปลข่าว METAR/SPECI และ TAF ของสนามบินที่เลือก ดังภาพที่ 5.12 ส่วนประกอบ และรายละเอียด เป็นเช่นเดียวกับสนามบินในพื้นที่ภาคใต้ฝั่งตะวันออก ในหัวข้อที่ 5.1 แต่ในส่วนของภาพเรดาร์ตรวจอากาศนั้น ผู้ใช้งานจะต้องตั้งค่าด้วยตนเอง เพื่อความยืดหยุ่นในการใช้งาน และสามารถเปลี่ยนไปใช้เรดาร์อื่นได้ เมื่อเรดาร์หลักที่ใช้งาน ขัดข้อง ทั้งนี้ จะสามารถแชร์ให้ผู้อื่นใช้งานอื่นๆ ใช้งานการตั้งค่าได้ด้วย โดยการตั้งค่า สามารถคลิกได้ที่ปุ่ม “ตั้งค่าเรดาร์” (อยู่ใต้แนวโน้มสภาวะอากาศ (ภายใน 2 ชั่วโมง)) ดังภาพที่ 5.13 หรือไปที่ “สนามบินอื่นๆ” → “เรดาร์ตรวจอากาศ” แล้วคลิกที่ปุ่มรูปเฟือง (Gear) ดังภาพที่ 5.14



ภาพที่ 5.12 แสดงหน้าเว็บไซต์ข่าวอากาศการบินสำหรับสนามบินอื่นๆ

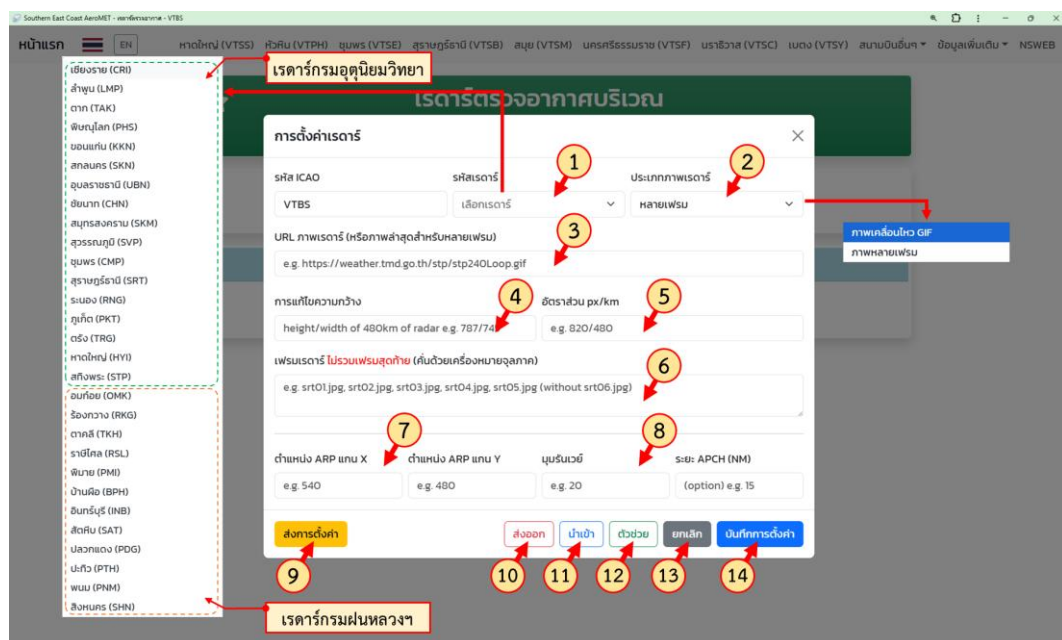


ภาพที่ 5.13 แสดงปุ่มตั้งค่าเรดาร์



ภาพที่ 5.14 แสดงหน้าเว็บไซต์เรดาร์ตรวจอากาศสำหรับสนามบินอื่นๆ

เมื่อคลิกที่ปุ่มแล้ว จะแสดงแบบฟอร์มการตั้งค่าดังภาพที่ 5.15 เพื่อกำหนดรายละเอียดต่างๆ



ภาพที่ 5.15 แสดงแบบฟอร์มการตั้งค่าภาพเรดาร์

ขั้นตอนการตั้งค่าภาพเรดาร์

1) เลือกเรดาร์ที่สนามบินใช้เป็นประจำ จาก <https://weather.tmd.go.th> ของกรมอุตุนิยมวิทยา หรือ https://www.royalrain.go.th/royalrain/sct_Radar.aspx ของกรมฝนหลวงและการบินเกษตร (อาจมีการเปลี่ยนแปลงในภายหลัง)

2) สำหรับเรดาร์ของกรมอุตุนิยมวิทยา ให้เลือกประเภทของภาพเรดาร์แบบเคลื่อนไหว (TMD Radar Loop) ให้เลือก “ภาพเคลื่อนไหว .GIF” หรือ “ภาพหลายเฟรม” โดยสังเกตจากลิงก์ของภาพเรดาร์นั้นๆ เช่น ภาพเรดาร์สทิงพระ มีลิงก์คือ “<https://weather.tmd.go.th/stp/stp240Loop.gif>” จะเป็นภาพเรดาร์แบบ “ภาพเคลื่อนไหว .GIF” และภาพเรดาร์ สุราษฎร์ธานี มีลิงก์คือ “<https://weather.tmd.go.th/srt/srt01.jpg>”, ..., “[.../srt/srt06.jpg](https://weather.tmd.go.th/srt/srt06.jpg)” จะเป็นภาพเรดาร์แบบ “ภาพหลายเฟรม” ส่วนเรดาร์ของกรมฝนหลวงฯ ไม่ต้องเลือกอะไร

3) กรอกลิงก์ของภาพเรดาร์ ถ้าเป็น “ภาพเคลื่อนไหว .GIF” → ให้ใช้ลิงก์ภาพเรดาร์แบบเคลื่อนไหว และถ้าเป็น “ภาพหลายเฟรม” → ให้ใช้ลิงก์ภาพเรดาร์ภาพล่าสุดของเรดาร์นั้น เช่น ใช้ลิงก์ “https://weather.tmd.go.th/srt/srt240_latest.jpg” เป็นต้น

4) แก้ไขความกว้าง เนื่องจากภาพเรดาร์บางสถานี จะถูกบีบในแนวนอน ทำให้ความกว้างของภาพเรดาร์ผิดเพี้ยน เช่น เรดาร์ลำพูน เรดาร์ภูเก็ต เรดาร์ชุมพร เรดาร์หาดใหญ่ เป็นต้น ถ้าหากสมมาตร จะมีค่าเป็น 1 และถ้าหากไม่สมมาตร ให้ระบุอัตราส่วนความสูงต่อความกว้างของภาพเรดาร์ (หน่วยพิกเซล: Pixel)

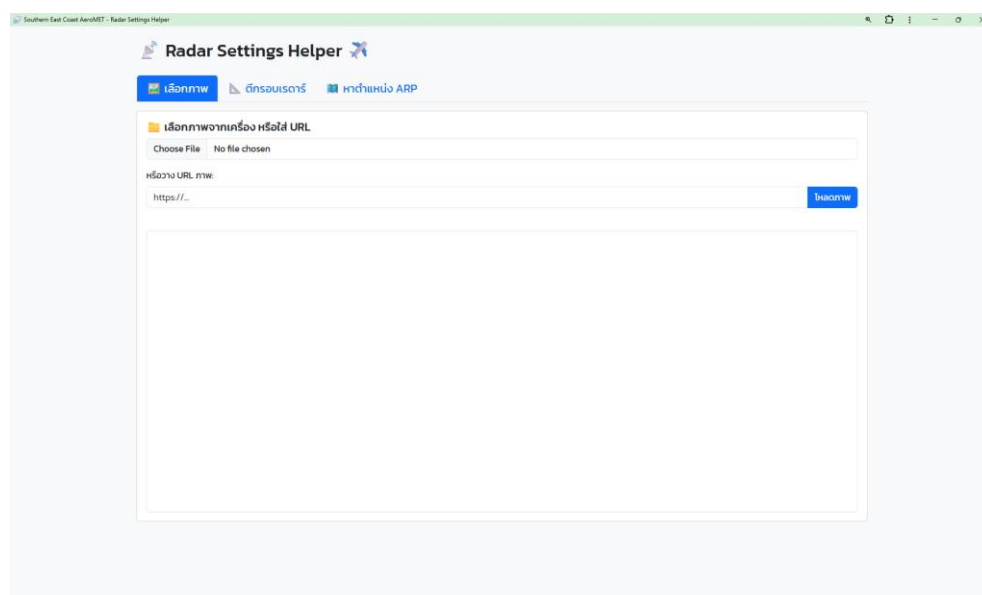
5) อัตราส่วนพิกเซลต่อกิโลเมตร ให้หาว่า เส้นวงกลมเส้นผ่านศูนย์กลาง 480 กิโลเมตร (หรืออื่นๆ) กว้างกี่พิกเซลในภาพเรดาร์นั้น

6) หากประเภทภาพเรดาร์เป็นแบบภาพหลายเฟรม ให้กรอกส่วนท้ายของลิงก์ ของภาพเฟรมอื่นๆ ยกเว้นเฟรมล่าสุด (เพราะระบุไว้ในขั้นตอนที่ 3 แล้ว) เช่น srt01.jpg, srt02.jpg, srt03.jpg, srt04.jpg, srt05.jpg (ไม่ต้องกรอก srt06.jpg)

7) หาตำแหน่งของจุด APR สนามบิน (X, Y) ในภาพเรดาร์ ตามพิกัดพิกเซล ถ้าหากมีการแก้ไขความกว้าง ตามขั้นตอนที่ 4 จะต้องปรับแก้ความกว้างนั้นก่อน แล้วหาพิกัดโดยอาจเปรียบเทียบกับแผนที่ใน Google Map

8) ระบุมุมของรันเวย์ โดยดูได้จาก <https://aip.caat.or.th> หัวข้อที่ 2.12 RUNWAY PHYSICAL CHARACTERISTICS ช่อง TRUE BRG และระบุระยะ Approach Control (หากทราบ)

ซึ่งขั้นตอนที่ 4 5 และ 7 สามารถใช้ตัวช่วย โดยคลิกที่ปุ่ม “ตัวช่วย” ดังภาพที่ 5.14 (จุดที่ 12) จะแสดงหน้าเว็บไซต์ดังภาพที่ 5.16 โดยจะให้เลือกภาพเรดาร์ที่ต้องการใช้งาน จากนั้นตีกรอบรอบเส้นวงกลมเส้นผ่านศูนย์กลาง 480 กิโลเมตร (หรืออื่นๆ) เว็บไซต์จะคำนวณการแก้ไขความกว้างและอัตราส่วนพิกเซลต่อกิโลเมตร ให้ จากนั้นให้คลิกบนภาพเรดาร์ เพื่อหาตำแหน่งของจุด APR สนามบิน (X, Y) หรือจะบันทึกเป็นรูปภาพ เพื่อนำไปหาในโปรแกรมอื่นๆ เพื่อความแม่นยำยิ่งขึ้น



ภาพที่ 5.16 แสดงหน้าเว็บไซต์ตัวช่วยตั้งค่าภาพเรดาร์

ปุ่มอื่นๆ

- ปุ่ม “ส่งการตั้งค่า” (จุดที่ 9) เมื่อคลิกแล้ว จะบันทึกการตั้งค่าภาพเรดาร์ของสนามบินนั้นๆ ไปที่เซิร์ฟเวอร์ เพื่อให้ผู้ใช้งานคนอื่นสามารถใช้งานการตั้งค่าได้ด้วย

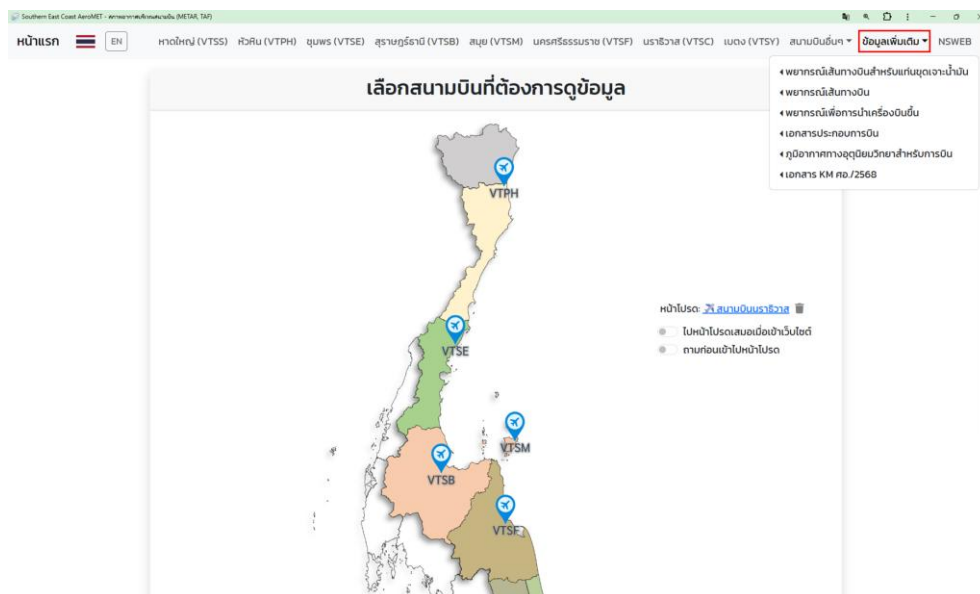
- ปุ่ม “ส่งออก” (จุดที่ 10) เมื่อคลิกแล้ว จะส่งออกไฟล์การตั้งค่าในรูปแบบ .json file เพื่อสำรองการตั้งค่า

- ปุ่ม “นำเข้า” (จุดที่ 11) เมื่อคลิกแล้ว จะให้นำเข้าไฟล์การตั้งค่า (.json) เพื่อกู้คืนการตั้งค่า ซึ่งการตั้งค่าจะยังไม่ถูกบันทึก ต้องคลิกที่ปุ่ม “บันทึกการตั้งค่า” หรือ “ส่งการตั้งค่า” ก่อน

- ปุ่ม “บันทึกการตั้งค่า” (จุดที่ 14) เมื่อคลิกแล้ว จะบันทึกการตั้งค่าไว้ที่เบราว์เซอร์ที่เปิดเว็บไซต์ในขณะนั้น (ถ้าเปิดเว็บไซต์ในอีกเว็บเบราว์เซอร์ จะไม่เจอการตั้งค่านี้) แล้วถ้าคลิกค้าง จะเป็นการลบการตั้งค่า แล้วจะไปใช้การตั้งค่าที่มีอยู่ในเซิร์ฟเวอร์แทน (ถ้ามี) ซึ่งเว็บไซต์ จะเลือกใช้ข้อมูลการตั้งค่าจากเว็บเบราว์เซอร์ก่อนเซิร์ฟเวอร์

5.3 ข้อมูลเพิ่มเติม

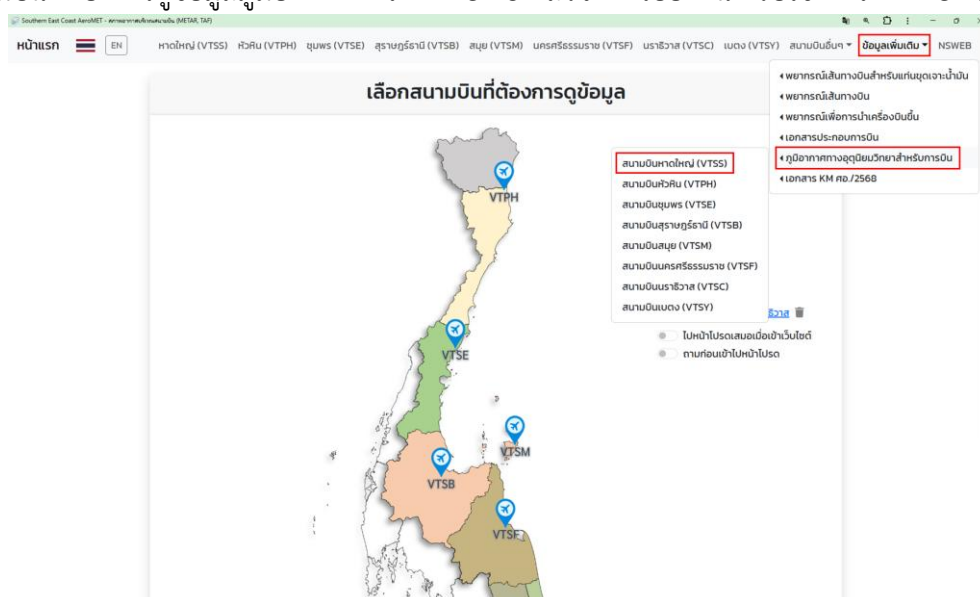
ประกอบไปด้วย พยากรณ์เส้นทางบินสำหรับแท่นขุดเจาะน้ำมัน, พยากรณ์เส้นทางบิน, พยากรณ์เพื่อการนำเครื่องบินขึ้น, เอกสารประกอบการบิน และ ภูมิอากาศทางอุตุนิยมวิทยาสำหรับการบิน ดังภาพที่ 5.17



ภาพที่ 5.17 แสดงข้อมูลเพิ่มเติมสำหรับอุตุนิยมวิทยาการบิน

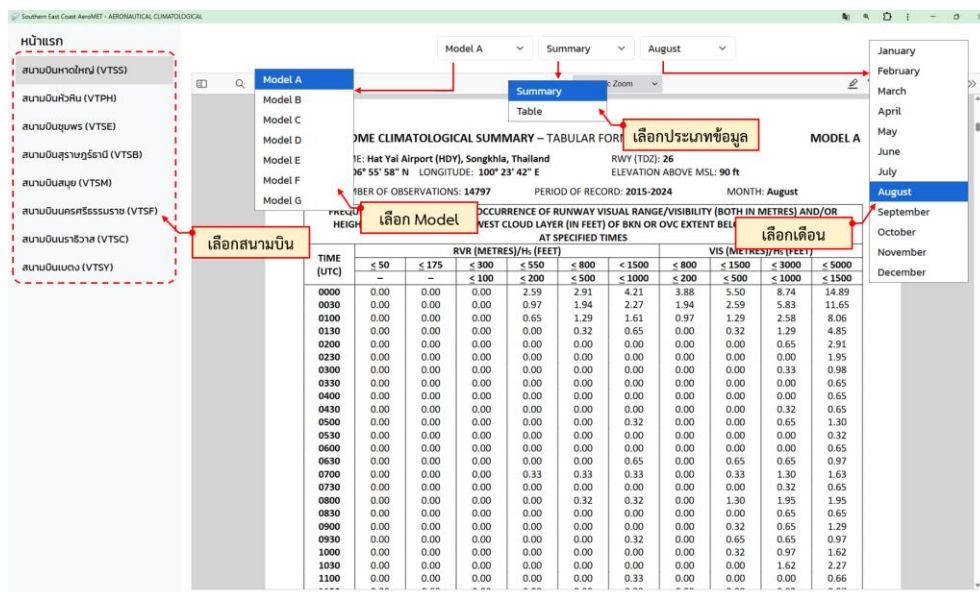
5.4 ภูมิอากาศสนามบิน (Aerodrome Climatological)

เข้าไปที่ ข้อมูลเพิ่มเติม → ภูมิอากาศทางอุตุนิยมวิทยาสำหรับการบิน → แล้วเลือกสนามบินที่ต้องการดูข้อมูลภูมิอากาศ ดังภาพที่ 5.18 แล้วจะพาไปยังหน้าเว็บไซต์ดังภาพที่ 5.19



ภาพที่ 5.18 แสดงขั้นตอนไปยังหน้าเว็บไซต์ภูมิอากาศสนามบิน

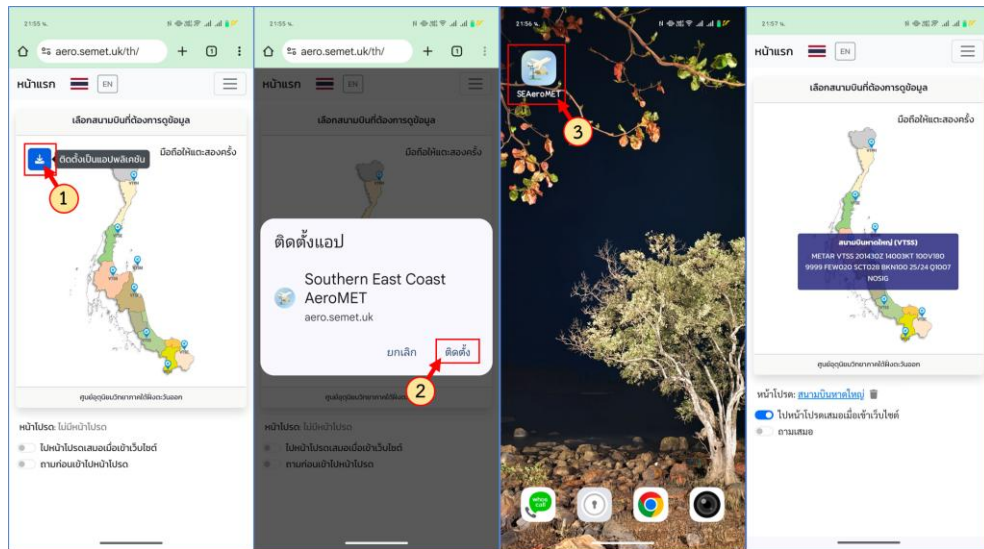
หน้าเว็บไซต์ภูมิอากาศทางอุตุนิยมวิทยาสำหรับการบิน จะมีให้เลือกสนามบินที่ต้องการดู ข้อมูลภูมิอากาศ สามารถเลือกดูโมเดล A ถึงโมเดล G, ประเภทข้อมูลแบบ Summary หรือ Table และเลือกเดือนที่ต้องการดูข้อมูลได้



ภาพที่ 5.19 แสดงหน้าเว็บไซต์ภูมิอากาศสนามบิน

5.5 การใช้งานบนโทรศัพท์มือถือ

หน้าเว็บไซต์จะถูกปรับให้อยู่ในอัตราส่วนที่เหมาะสมสำหรับโทรศัพท์มือถือ แต่การใช้งานยังคงเหมือนเดิม สามารถใช้งานเว็บไซต์ได้บนเว็บเบราว์เซอร์ หรือจะติดตั้งเป็นแอปพลิเคชันแบบ PWA ก็ได้ โดยจะต้องเปิดเว็บไซต์ <https://aero.semet.uk> บนเว็บเบราว์เซอร์ที่รองรับการติดตั้งแบบ PWA เช่น Chrome, Microsoft Edge, Safari เป็นต้น (แนะนำ Chrome) โดยแตะสองครั้งที่ปุ่ม “ติดตั้งเป็นแอปพลิเคชัน” ดังภาพที่ 5.20(ก) หากปุ่มนี้ไม่ปรากฏให้เห็น ให้เพิ่มไปยังหน้าจอหลัก หรือ Add to Home Screen ด้วยตัวเอง ตามวิธีของแต่ละเว็บเบราว์เซอร์ เมื่อทำการติดตั้งแล้ว จะปรากฏเป็นแอปพลิเคชันบนหน้าจอ



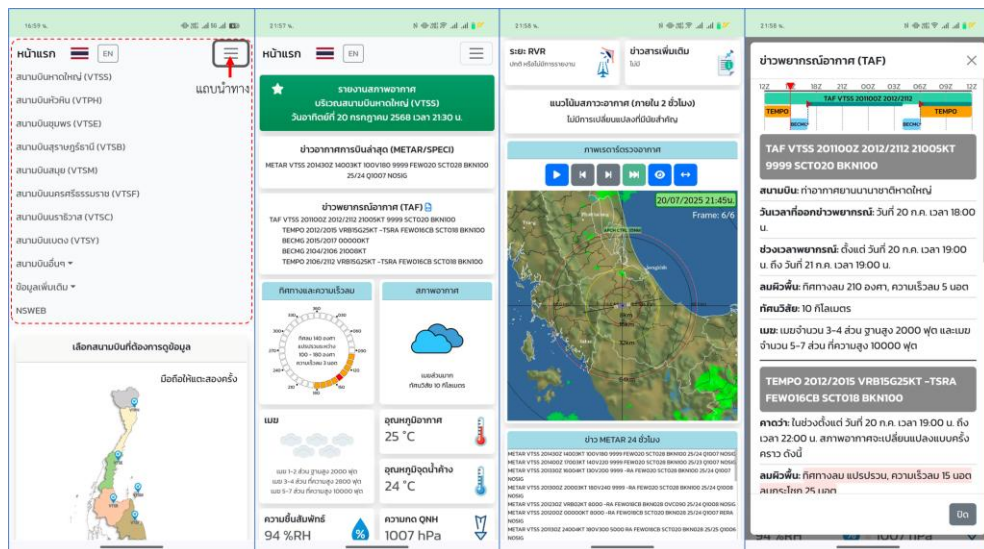
(ก)

(ข)

(ค)

(ง)

ภาพที่ 5.20 แสดงการติดตั้งเว็บไซต์เป็นแอปพลิเคชันแบบ PWA



(ก)

(ข)

(ค)

(ง)

ภาพที่ 5.21 แสดงการใช้งานเว็บไซต์บนโทรศัพท์มือถือ

บรรณานุกรม

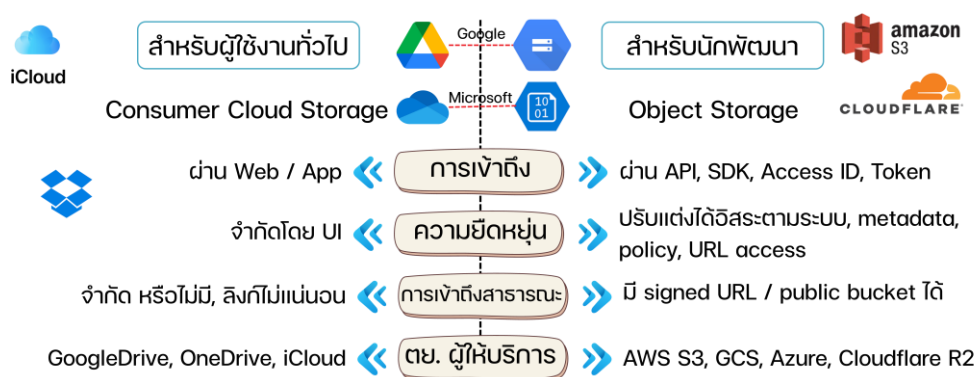
- กองอุตุนิยมวิทยาการบิน. 2563. คู่มือมาตรฐาน การตรวจและรายงานอากาศการบินแบบ METAR และ SPECI. กรมอุตุนิยมวิทยา
- ณัฐกิตติ์ งามหมู่. 2568. ซอฟต์แวร์ช่วยปฏิบัติงานด้านอุตุนิยมวิทยาการบิน และการเผยแพร่ข้อมูลอุตุนิยมวิทยาของสนามบินนราธิวาสทางเว็บไซต์. กรมอุตุนิยมวิทยา
- สุรศักดิ์ แซ่ลิ้ม. “Hello World, I am AI.”: AI คืออะไร ครูไทยต้องรู้ !. เข้าถึงได้ที่ <https://www.educathai.com/knowledge/articles/668>
- ส่วนมาตรฐานอุตุนิยมวิทยาการบิน. 2565. คู่มือมาตรฐาน การพยากรณ์แนวโน้มสภาวะอากาศบริเวณสนามบิน. กองอุตุนิยมวิทยาการบิน. กรมอุตุนิยมวิทยา
- ส่วนมาตรฐานอุตุนิยมวิทยาการบิน. 2565. คู่มือมาตรฐาน การพยากรณ์อากาศบริเวณสนามบิน. กองอุตุนิยมวิทยาการบิน. กรมอุตุนิยมวิทยา
- Borntodev. 2020. 10 เรื่องน่ารู้ตอบข้อสงสัย JavaScript. เข้าถึงได้ที่ <https://www.borntodev.com/2020/01/06/10-เรื่องน่ารู้-javascript>
- Code-th.com. ตกแต่งด้วย CSS. เข้าถึงได้ที่ <https://code-th.com/html/lesson/css>
- Dynamic Intelligence Asia (dIA). ปัญญาประดิษฐ์ (AI) คืออะไร หลักการทำงาน และการใช้ในอุตสาหกรรม. เข้าถึงได้ที่ <https://www.dia.co.th/articles/what-is-artificial-intelligence>
- DATAHOLIC TEAM. Landscape ของ Generative AI มีกี่ประเภท ?. เข้าถึงได้ที่ <https://dataholictth.com/info/generative-ai-landscape>
- Enfete. 2024. ภาษา HTML และโครงสร้างของภาษา HTML เพื่อเริ่มต้นเขียนเว็บไซต์ด้วยตัวเอง เข้าถึงได้ที่ <https://enfete.co.th/th/html-language-and-html-structure>
- International Civil Aviation Organization (ICAO). 2018. Annex 3 Meteorological Service for International Air Navigation, 20th ed. Montreal, Canada.
- KongRuksiam Studio. 2024. พื้นฐาน JavaScript สำหรับการพัฒนาเว็บแอปพลิเคชัน (HTML & CSS). เข้าถึงได้ที่ <https://kongruksiam.medium.com/พื้นฐาน-javascript-สำหรับการพัฒนาเว็บแอปพลิเคชัน-html-css-1ff1e34dc2af>
- Thanatcha, V. 2022. FRONT END, BACK END, FULL STACK คืออะไร มีความแตกต่างกันอย่างไรบ้าง ?. เข้าถึงได้ที่ <https://blog.openlandscape.cloud/front-end>

ภาคผนวก

ภาคผนวก ก.

แนะนำผู้รับบริการสำหรับจัดเก็บข้อมูลบนคลาวด์ (Cloud Storage)

Cloud Storage คือ พื้นที่จัดเก็บข้อมูลบนคลาวด์ ซึ่งเป็นบริการที่อนุญาตให้ผู้ใช้สามารถเก็บข้อมูลหรือไฟล์ต่าง ๆ ไว้บนเซิร์ฟเวอร์ออนไลน์ สามารถเข้าถึงได้ผ่านอินเทอร์เน็ต การใช้ Cloud Storage นอกจากจะช่วยให้สามารถเข้าถึงข้อมูลได้จากทุกที่และทุกอุปกรณ์แล้ว ยังช่วยให้การจัดการข้อมูลมีความปลอดภัยและมีประสิทธิภาพมากขึ้น การเปรียบเทียบ Cloud Storage สำหรับผู้ใช้งานทั่วไป และสำหรับนักพัฒนา แสดงดังภาพที่ ก.1



ภาพที่ ก.1 แสดงการเปรียบเทียบ Cloud Storage ระหว่าง
สำหรับผู้ใช้งานทั่วไป และสำหรับนักพัฒนา

Object Storage (การจัดเก็บในรูปแบบวัตถุ) เป็นระบบจัดเก็บข้อมูลที่ออกแบบมาเพื่อรองรับข้อมูลขนาดใหญ่ประเภทไม่มีโครงสร้าง (Unstructured Data) เช่น ไฟล์วิดีโอ เอกสาร และอีเมล โดยเก็บข้อมูลเป็นวัตถุที่ประกอบด้วยเนื้อหา (Data) ข้อมูลเมตา (Metadata) และรหัสระบุเฉพาะ (Unique Identifier: UID) ระบบนี้สามารถขยายตัวได้อย่างไร้ขอบเขตและเข้าถึงข้อมูลผ่าน RESTful API, HTTP หรือ HTTPS จึงเหมาะสำหรับงานคลาวด์ที่ต้องการความยืดหยุ่นและต้นทุนต่ำ ตัวอย่างผู้ให้บริการ เช่น Amazon S3, Google Cloud Storage, Azure Blob Storage, Oracle Object Storage และ Cloudflare R2 เป็นต้น

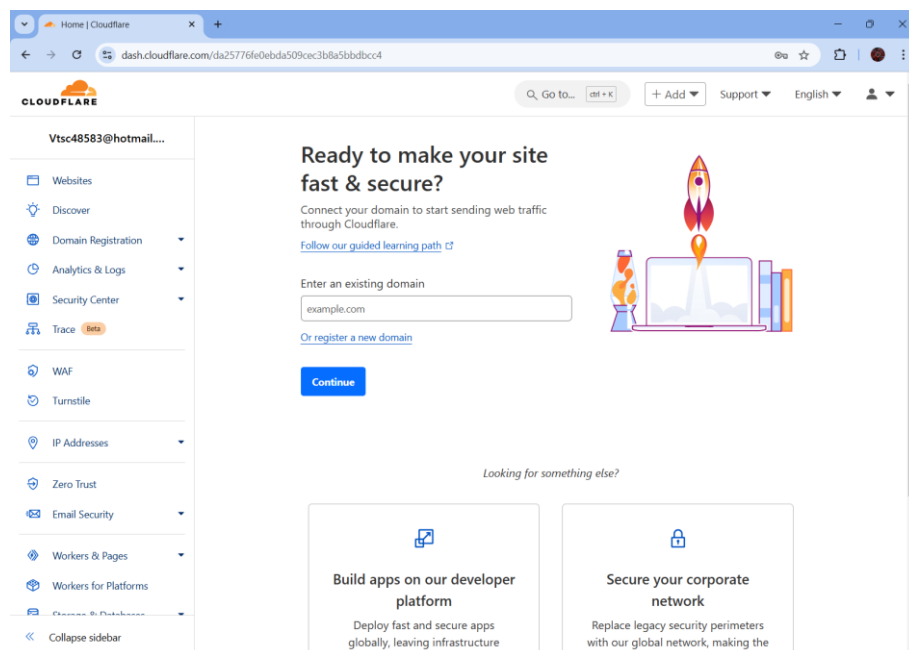
Cloudflare R2 เป็นหนึ่งใน Cloud Storage ประเภท Object Storage ที่ออกแบบมาเพื่อใช้งานสำหรับ นักพัฒนา หรือ องค์กร ที่ต้องการเก็บไฟล์ในระดับขนาดใหญ่และเข้าถึงผ่าน API สามารถเข้าถึงไฟล์ภายในได้ ทั้งกรณีที่ให้สิทธิ์เข้าถึงเฉพาะบางคน หรือแบบสาธารณะ (Public link) และลิงก์นั้นสามารถนำไปใช้งานได้โดยตรง มีข้อดีคือ ไม่มีค่าทำเนียมการรับส่งข้อมูล (Free Ingress & Egress Fees) แต่ขั้นตอนการเปิดใช้งาน จะต้องผูกกับบัตรเครดิตหรือบัตรเครดิต (ใช้บัตรเสมือนได้)

ข้อจำกัดเมื่อใช้งานฟรี

- 1) จัดเก็บได้ 10 GB ต่อเดือน
- 2) การดำเนินการ Class A (ListBuckets, ListObjects, PutObject, ...) รวมถึงการอัปโหลดไฟล์ 1 ล้านครั้งต่อเดือน
- 3) การดำเนินการ Class B (HeadBucket, HeadObject, GetObject, ...) รวมถึงการดาวน์โหลดไฟล์ เรียกดูไฟล์ 10 ล้านครั้งต่อเดือน

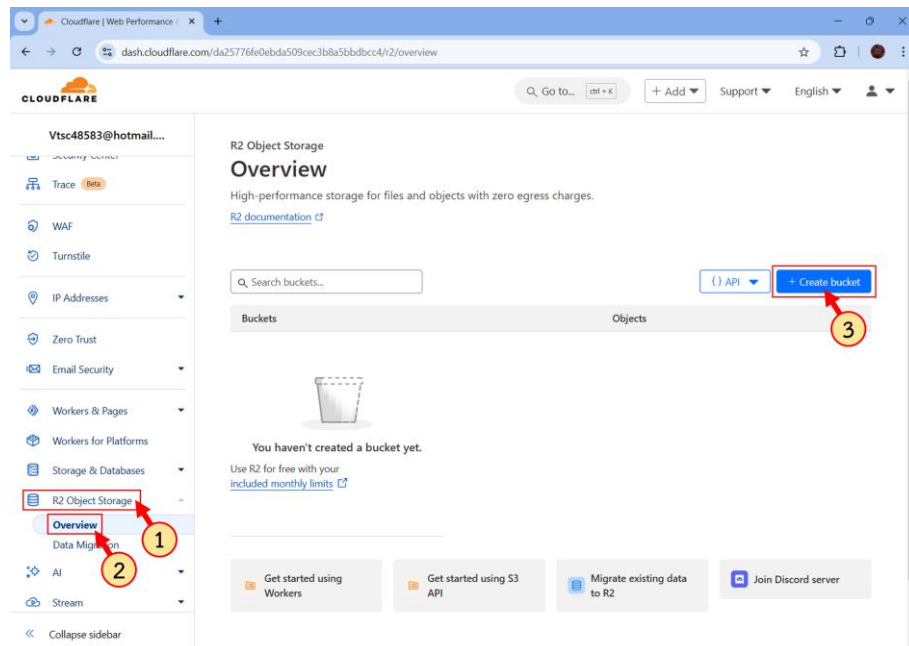
ขั้นตอนการดำเนินการ

- 1) สมัครใช้งานที่เว็บไซต์ <https://dash.cloudflare.com>



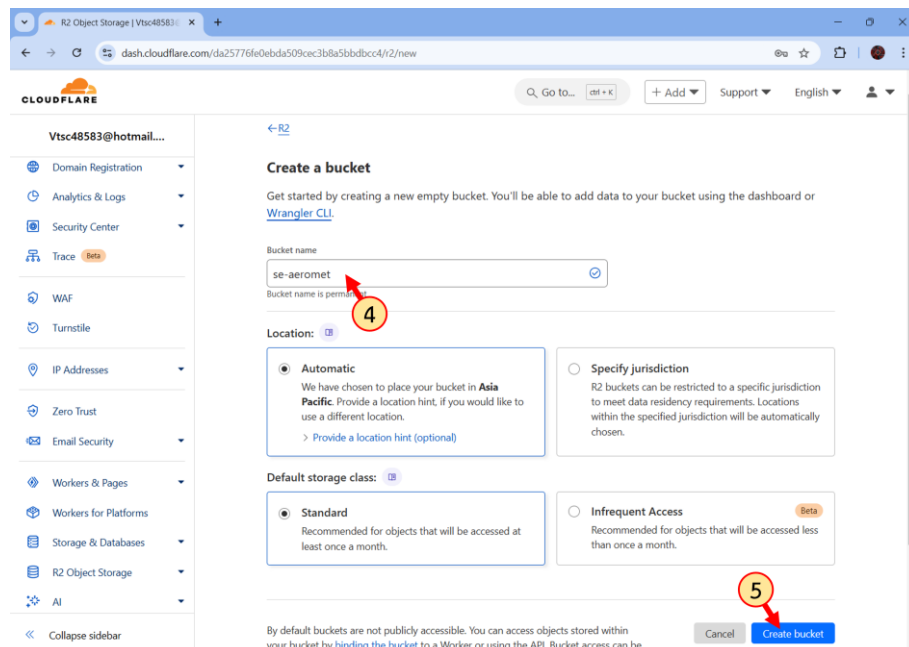
ภาพที่ ก.2 แสดงเว็บไซต์ cloudflare.com

2) ไปที่ R2 Object Storage -> Overview แล้วคลิก Create Bucket



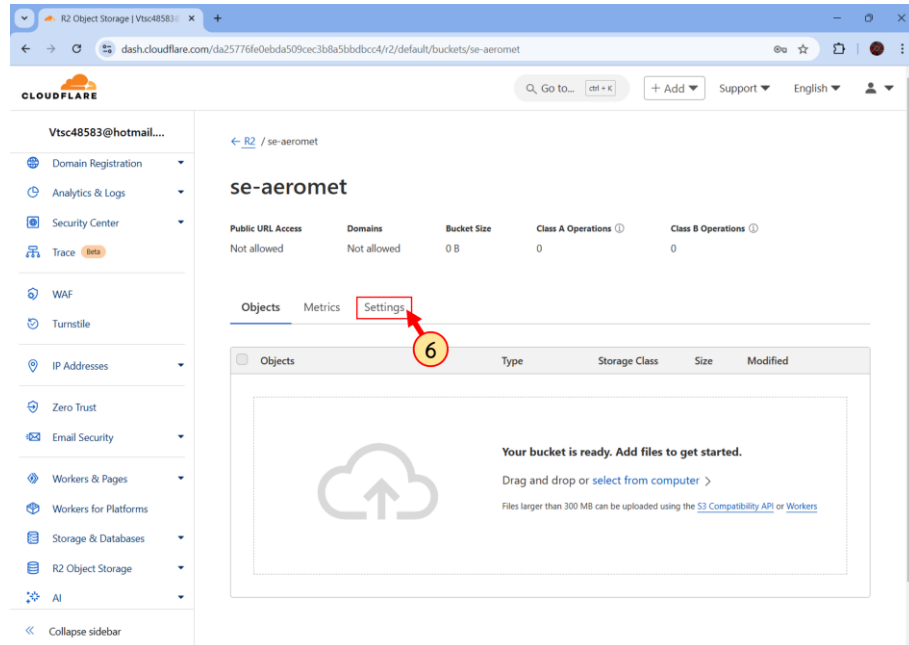
ภาพที่ ก.3 แสดงขั้นตอนการสร้าง Bucket

3) ตั้งชื่อ Bucket แล้วคลิก Create Bucket



ภาพที่ ก.4 แสดงขั้นตอนการสร้าง Bucket (ต่อ)

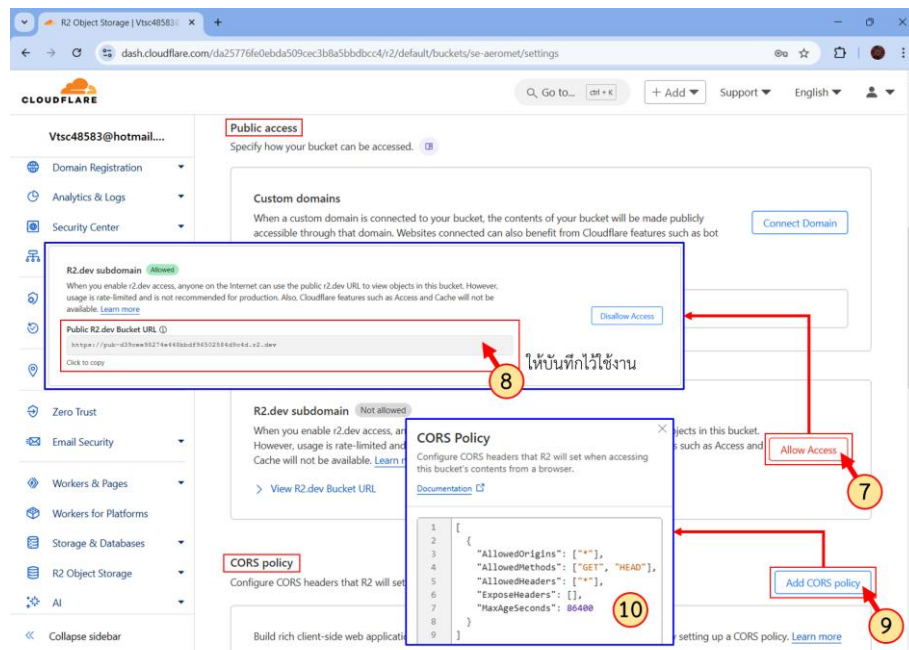
4) สร้าง Bucket เรียบร้อย พร้อมให้อัพโหลดไฟล์ แต่ก่อนที่เราจะเผยแพร่ในครั้งแรก ต้องตั้งค่า Bucket ให้ เป็น Public ก่อน ให้คลิกที่ Setting



ภาพที่ ก.5 แสดงการตั้งค่า Bucket

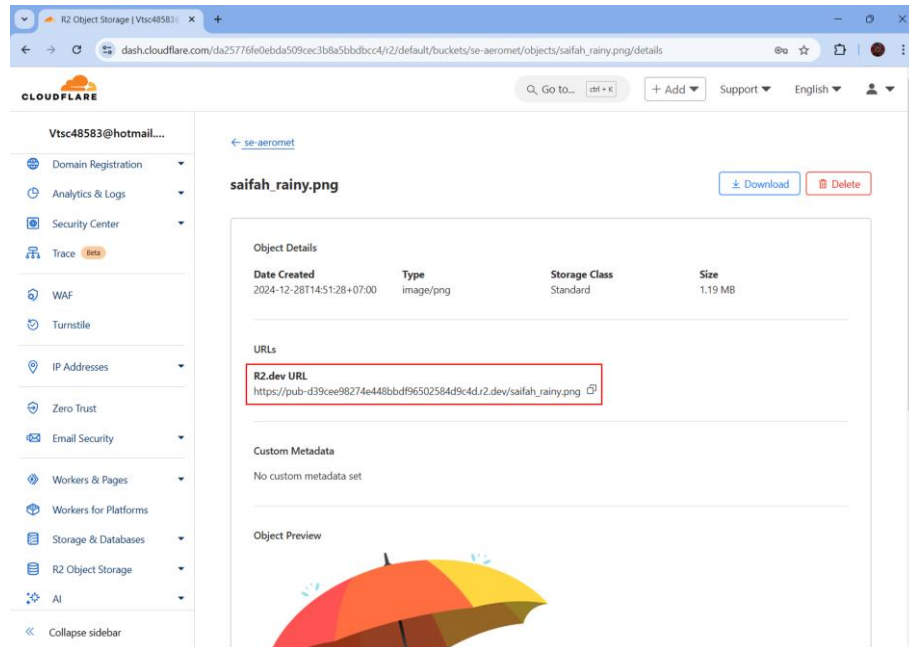
5) เลื่อนไปที่ Public access → Allow Access จะได้ **Public R2.dev Bucket** URL ให้บันทึกไว้ใช้งาน และเพื่อให้เข้าถึงไฟล์ใน Bucket ได้อย่างเต็มที่ ให้เพิ่ม CORS policy → add CORS policy ดังภาพที่ ก.6

```
[
  {
    "AllowedOrigins": ["*"],
    "AllowedMethods": ["GET", "HEAD"],
    "AllowedHeaders": ["*"],
    "ExposeHeaders": [],
    "MaxAgeSeconds": 86400
  }
]
```



ภาพที่ ก.6 แสดงการตั้งค่า Bucket

6) เมื่อทำการอัปโหลดไฟล์แล้ว เราจะได้ R2.dev URL มา สามารถนำลิงก์นั้นไปใช้งานได้เลย จะเห็นว่าลิงก์เป็น Public R2.dev Bucket URL + ชื่อไฟล์ที่เราอัปโหลด



ภาพที่ ก.7 แสดงลิงก์ของไฟล์ที่ถูกอัปโหลด

ขั้นตอนการอัปโหลดไฟล์นั้น สามารถทำได้ด้วยการอัปโหลดตรงๆ ที่หน้าเว็บ Cloudflare ได้เลย (Drag & Drop) แต่การอัปโหลดด้วยวิธีนี้ จะมีข้อจำกัด ไม่ควรใช้กับไฟล์ที่มีการอัปเดตบ่อย เช่น ไฟล์ภาพอินโฟกราฟิกพยากรณ์อากาศที่อัปเดตเป็นประจำทุกวัน หรือไฟล์ภาพเรดาร์ที่อัปเดตทุกๆ 15 นาที ข้อจำกัดคือ ไม่รองรับไฟล์ที่มีขนาดใหญ่ (>5GB), การแสดงผลของไฟล์บนเว็บเบราว์เซอร์อาจจะผิดเพี้ยน เนื่องจากไม่ได้กำหนดรูปแบบของไฟล์ (Mime Type) และที่สำคัญไม่ได้กำหนด Cache-Control อาจทำให้ไฟล์ถูกแคชไว้ที่เว็บเบราว์เซอร์ แล้วเจอปัญหาไฟล์ไม่อัปเดต แม้จะอัปโหลดไฟล์ใหม่แล้วก็ตาม เช่น อัปโหลดภาพอินโฟกราฟิกภาพใหม่แล้ว แต่ยังคงแสดงบนเว็บไซต์เป็นภาพเก่าอยู่ (เนื่องจากภาพเก่าถูกแคชไว้ที่เว็บเบราว์เซอร์) จึงแนะนำให้อัปโหลดไฟล์ผ่าน API

การอัปโหลดไฟล์ผ่าน API มีขั้นตอนดังนี้

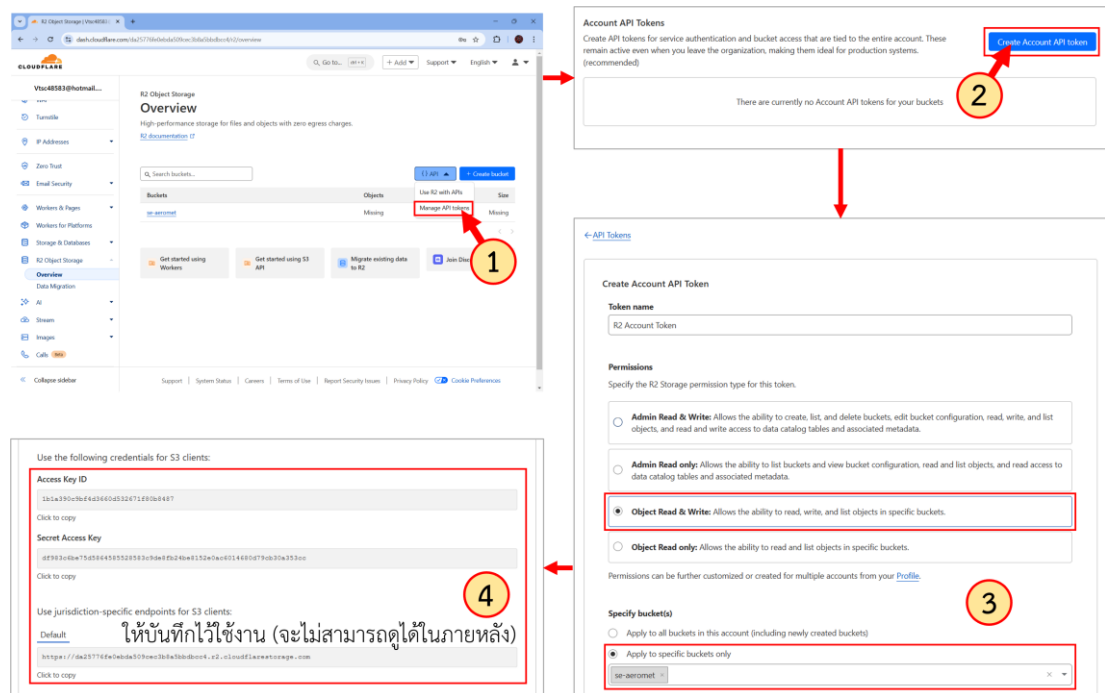
1) ในครั้งแรกต้องขอ API Token ก่อน เมื่อได้ Access Key ID, Secret Key ID และ endpoint for S3 มาแล้ว ให้คัดลอกเก็บไว้ใช้ในการอัปโหลดไฟล์ แสดงดังภาพที่ ก.8

(1) คลิกที่ {}API → Manage API tokens

(2) Create Account API Token

(3) ตั้งค่า การอนุญาตการเข้าถึง Bucket แนะนำให้เลือก Object Read & Write และแนะนำให้เลือกเฉพาะ Bucket ที่ใช้งาน (Apply to specific buckets only) ขั้นตอนนี้ สามารถกลับมาเปลี่ยนการตั้งค่าในภายหลังได้

(4) จะได้ Access Key ID, Secret Access Key และ endpoint for S3 clients ให้บันทึกเก็บไว้ เพราะจะไม่สามารถกลับมาได้ในภายหลัง



ภาพที่ ก.8 แสดงขั้นตอนการขอ API Token

2) ตัวอย่างการอัปโหลดไฟล์โดยใช้ภาษาไพธอน (Python) โดยจะต้องติดตั้งไลบรารี boto3 (ใช้คำสั่ง “pip install boto3”) เป็น AWS SDK สำหรับ Python ของ Amazon Simple Storage Service (S3) ซึ่งรองรับการใช้งาน (Compatible) กับ Cloudflare R2 และใช้ไลบรารี mimetypes เพื่อกำหนด Mime Type คือมาตรฐานการสื่อสารของไฟล์เอกสารต่างๆ เพื่อระบุให้เว็บเบราว์เซอร์รู้ว่าไฟล์ที่กำลังติดต่อกับเซิร์ฟเวอร์นั้นคือไฟล์รูปแบบใด (โดยการกำหนด HTTP Header: Content-Type) เช่น ไฟล์ภาพ PNG, JPEG, JSON, หรือไฟล์ PDF ทำให้การแสดงผล หรือการใช้งานไฟล์นั้นๆ ได้ถูกต้อง และที่สำคัญคือการระบุ HTTP Header: Cache-Control ซึ่งมีความสำคัญในการควบคุมการเก็บแคชของข้อมูลในเว็บเบราว์เซอร์ หรือใน CDN ช่วยให้ควบคุมว่า ข้อมูล หรือไฟล์ ที่ส่งจากเซิร์ฟเวอร์ควรจะถูกเก็บในแคช (cache) นานแค่ไหน หรือควรจะถูกโหลดใหม่ทุกครั้งหรือไม่ กำหนดได้หลายแบบ ตัวอย่างเช่น

- no-store : ไม่ให้เก็บข้อมูลนี้ในแคชเลย จะขอใหม่ตลอด → ข้อมูลที่ละเอียดอ่อน
- no-cache : ไม่ให้แคชข้อมูลนี้โดยตรง คือ มีการเก็บแคช แต่จะตรวจสอบกับเซิร์ฟเวอร์ก่อน ว่ามีการอัปเดตใหม่ไหม ถ้ามีการอัปเดต → โหลดใหม่ / ถ้าไม่มี ใช้ข้อมูลในแคชได้เลย เหมาะกับไฟล์ที่มีการอัปเดตบ่อยๆ เช่น ภาพเรดาร์ตรวจอากาศ
- max-age=<seconds> : กำหนดเวลาเป็นวินาที จะไม่มีการร้องขอไฟล์ใหม่จากเซิร์ฟเวอร์เลย จนกว่าแคชจะหมดอายุ เหมาะกับไฟล์ที่อัปเดตนานๆ ครั้ง
- ถ้าไม่กำหนด จะตั้งตามผู้ให้บริการ/ตามลักษณะไฟล์ อาจจะ 1 นาที หรือ 1 สัปดาห์ก็ได้ ซึ่งไม่เหมาะกับการใช้งานจริง

```

import boto3 # pip install boto3
import mimetypes # มีใน python อยู่แล้ว ไม่ต้องลงเพิ่ม

s3 = boto3.client(
    service_name="s3",
    endpoint_url="https://r2.cloudflarestorage.com",
    aws_access_key_id="...",
    aws_secret_access_key="...",
    region_name='auto'
)

def uploadfile(localfile, bucket, uploadedfile):
    with open(localfile, "rb") as f:
        s3.upload_fileobj(
            Fileobj=f,
            Bucket=bucket,
            Key=uploadedfile,
            ExtraArgs={
                "ContentType": mimetypes.guess_type(uploadedfile)[0],
                "CacheControl": "no-cache",
            },
        )

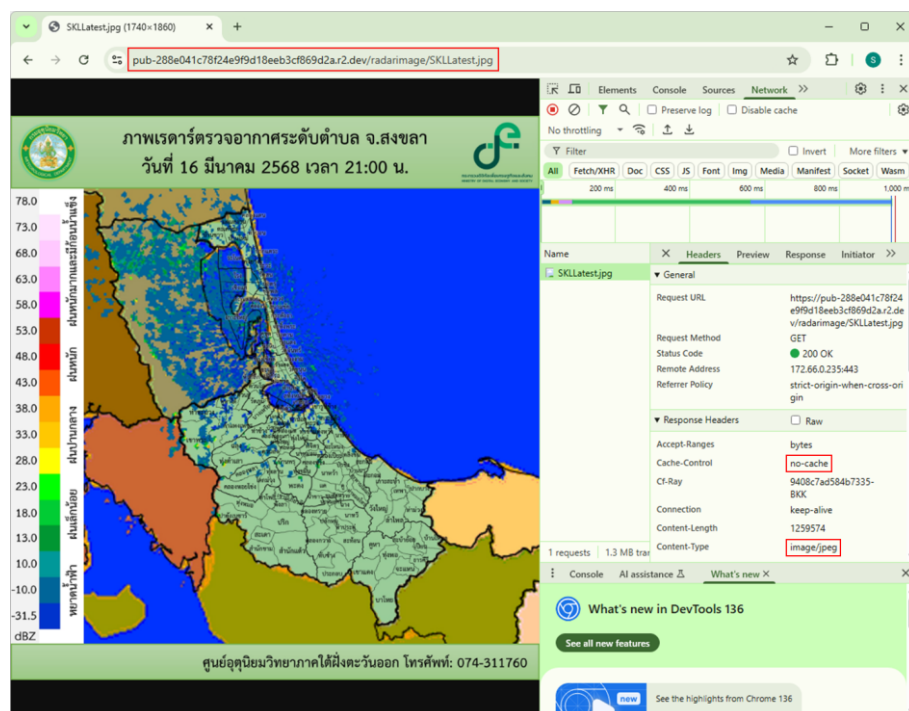
uploadfile('file/SKLLatest.jpg', 'se-aeromet', 'radarimage/SKLLatest.png')

```

ไฟล์ที่จะอัปโหลด ชื่อ Bucket ตำแหน่งไฟล์ปลายทาง

ภาพที่ ก.9 แสดงตัวอย่างโค้ดที่ใช้อัปโหลดไฟล์ โดยใช้ภาษา Python

3) สามารถเข้าถึงไฟล์ได้ที่ลิงก์ Public R2.dev Bucket URL + ตำแหน่งไฟล์ปลายทางที่เราอัปโหลด ซึ่งเราอาจนำไปประยุกต์ใช้กับ Line Messaging API ได้



ภาพที่ ก.10 แสดงตัวอย่างการเรียกดูรูปภาพที่ถูกอัปโหลด

ภาคผนวก ข.

การสร้างตัวกลาง (Proxy) สำหรับแก้ปัญหา CORS ในเว็บไซต์

เว็บไซต์ที่ถูกพัฒนาแบบไม่มีเซิร์ฟเวอร์ (Serverless Website) ทำให้การเข้าถึงข้อมูลจากเว็บไซต์ภายนอก เช่น การร้องขอ Application Programming Interface (API) ข่าว METAR/TAf จากเว็บไซต์ <https://aviationweather.gov> หรือการดึงข่าวจาก <https://news.tmd.go.th/#showMetars> หรือการดึงภาพเรดาร์จาก <https://weather.tmd.go.th> นั้น ถูกจำกัดด้วย การแบ่งปันทรัพยากรแบบข้ามต้นทาง (Cross-Origin Resource Sharing: CORS) ทำให้ต้องหาวิธีเพิ่มเติม ในการสร้างตัวกลางระหว่างเซิร์ฟเวอร์ (Server) และผู้ใช้งาน (Client) เช่น เว็บเบราว์เซอร์ เรียกว่า พร็อกซีเซิร์ฟเวอร์ (Proxy server) เพื่อเข้าถึงข้อมูลจากเซิร์ฟเวอร์ภายนอกนั้น ๆ

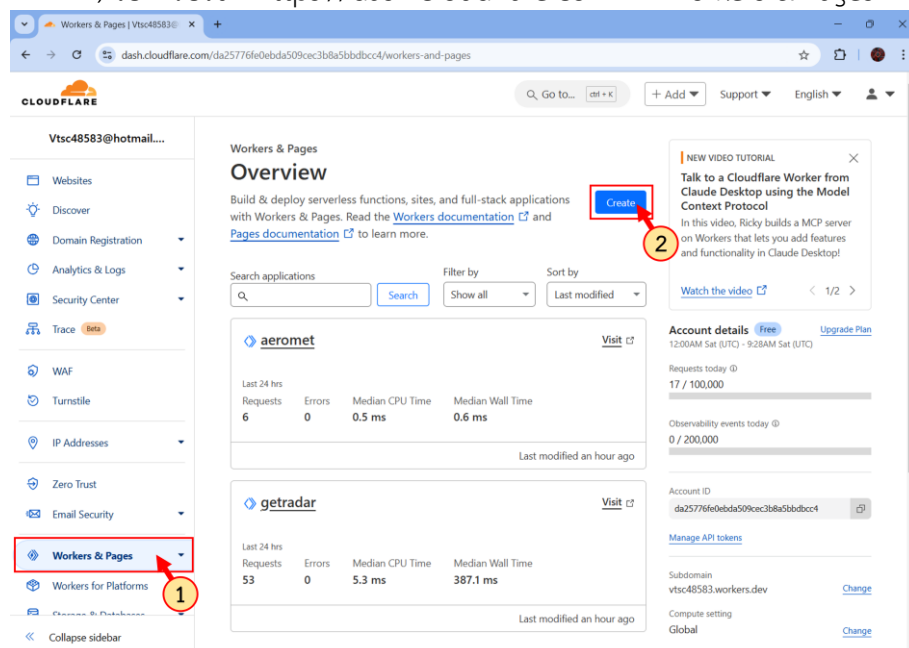
* หมายเหตุ หากนำภาพไปแสดงบนเว็บไซต์เฉยๆ ไม่จำเป็นต้องแก้ไข CORS สามารถนำไปแสดงได้เลย แต่ถ้าหากมีการอ่านข้อมูล เช่น แยกภาพ Gif ออกเป็นเฟรม ผสานภาพ หรืออ่านข้อมูลของไฟล์ text, json หรืออื่นๆ แล้วนำไปประมวลผล จำเป็นต้องทำการแก้ไข CORS ก่อน และบางผู้ให้บริการ ไม่อนุญาตให้ Client เรียก API โดยตรง ต้องศึกษา Terms of Use ของ API ก่อนใช้งาน

✖ Access to fetch at '<https://aviationweather.gov/api/data/metar?ids=VTSS&format=raw&taf=true&hours=24>' from origin 'http://127.0.0.1:5500' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.

ภาพที่ ข.1 แสดงการติดปัญหา CORS เมื่อร้องขอ API

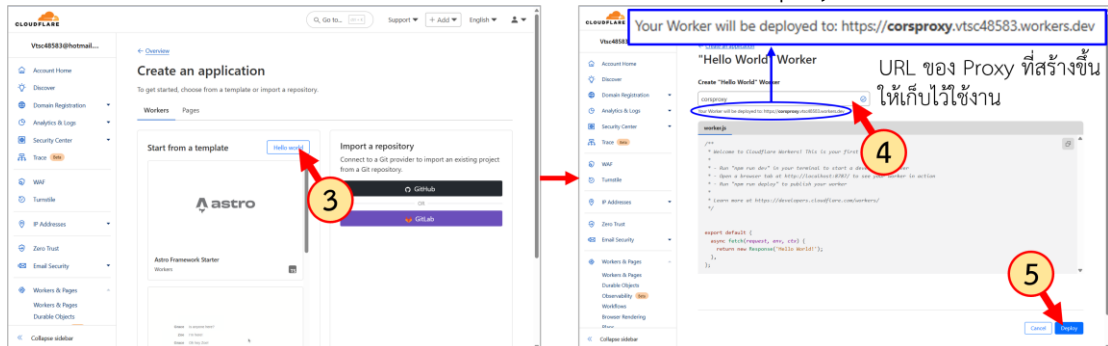
ขั้นตอนการดำเนินการ

1) ไปที่เว็บไซต์ <https://dash.cloudflare.com> → Workers & Pages → Create



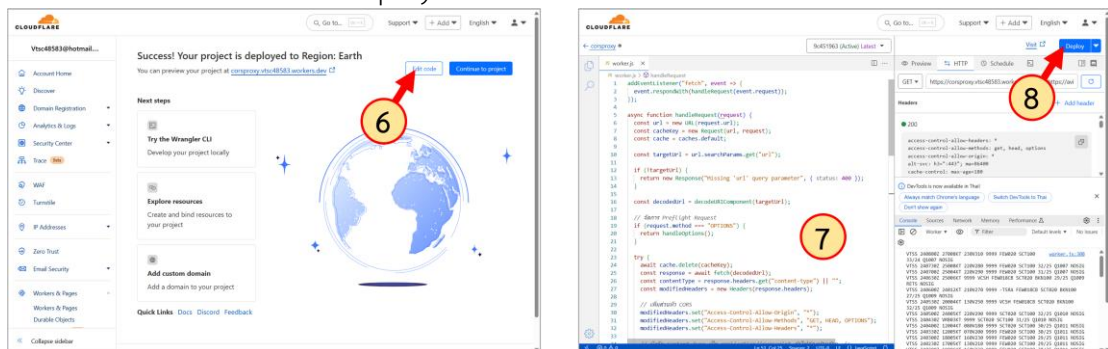
ภาพที่ ข.2 แสดงการสร้าง Proxy Server โดยใช้ Cloudflare Workers

2) คลิกที่ Hello world → ตั้งชื่อ Worker แล้วจะเห็นลิงก์ของ Worker ปรากฏด้านล่างช่องตั้งชื่อ ให้บันทึกไว้ เพื่อนำไปเรียกใช้งาน จากนั้นคลิกที่ Deploy



ภาพที่ ข.3 แสดงการสร้าง Proxy Server โดยใช้ Cloudflare Workers (ต่อ)

3) คลิกที่ Edit Code จากนั้น ให้พิมพ์โค้ดที่แก้ปัญหา CORS ลงไปที่ช่องพิมพ์โค้ด เมื่อทดสอบผ่านแล้ว ให้คลิกที่ Deploy ก็สามารถนำลิงก์ของ Worker ไปใช้งานได้



ภาพที่ ข.4 แสดงการสร้าง Proxy Server โดยใช้ Cloudflare Workers (ต่อ)

ตัวอย่างการใช้งาน

ถ้าหากลิงก์ของข้อมูลที่ต้องการแก้ไขปัญห CORS มี Query string (?) ต้องทำการ encodeURIComponent ก่อน ไม่งั้นนั้น Cloudflare Workers จะสับสนว่าเป็น Query string ของตัวเอง (จริงๆเป็นของลิงก์ต้นฉบับ) หากไม่มี สามารถนำลิงก์ไปต่อท้าย Proxy ที่สร้างได้เลย (ต่อท้าย https://...workers.dev/?url=) **ใช้ https://aviationweather.gov เพื่อการศึกษาเท่านั้น

```
<!DOCTYPE html>
<html lang="th">
<body>
  <p>ข่าว METAR ย้อนหลัง 24 ชั่วโมง</p>
  <p id="metar"></p>
  <script>
    document.addEventListener('DOMContentLoaded', async () => {
      const metarElement = document.getElementById('metar');
      const metarparam = '?ids=VTSS&format=raw&taf=true&hours=24';
      const metarUrl = 'https://aviationweather.gov/api/data/metar' + encodeURIComponent(metarparam);
      const response = await fetch('https://corsproxy.vtsc48583.workers.dev/?url=' + metarUrl);
      const data = await response.text();
      metarElement.innerText = data;
    });
  </script>
</body>
</html>
```

ภาพที่ ข.5 แสดงตัวอย่างการใช้งาน Proxy Server โดยใช้ Cloudflare Workers

ตัวอย่างโค้ดสำหรับสร้าง Proxy Server โดยใช้ Cloudflare Workers เพื่อใช้แก้ไขปัญหา CORS

```
addEventListener("fetch", event => {
  event.respondWith(handleRequest(event));
});
async function handleRequest(event) {
  const request = event.request;
  const url = new URL(request.url);
  const targetUrl = url.searchParams.get("url");
  if (!targetUrl) {
    return new Response("Missing 'url' query parameter", { status:
400 });
  }
  if (request.method === "OPTIONS") {
    return handleOptions();
  }
  const decodedUrl = decodeURIComponent(targetUrl);
  const cache = caches.default;
  const cacheKey = new Request(request.url, request);
  const cachedResponse = await cache.match(cacheKey);
  if (cachedResponse) {
    return cachedResponse;
  }
  try {
    const response = await fetch(decodedUrl);
    const headers = new Headers(response.headers);
    headers.set("Access-Control-Allow-Origin", "*");
    headers.set("Access-Control-Allow-Methods", "GET, HEAD,
OPTIONS");
    headers.set("Access-Control-Allow-Headers", "*");
    headers.set("Cache-Control", "public, max-age=0, s-maxage=60,
must-revalidate");
    const newResponse = new Response(response.body, {
      status: response.status,
      statusText: response.statusText,
      headers,
    });
    event.waitUntil(cache.put(cacheKey, newResponse.clone()));
    return newResponse;
  } catch (error) {
    return new Response(`Error: ${error.message}`, { status: 500 });
  }
}
function handleOptions() {
  const headers = new Headers();
  headers.set("Access-Control-Allow-Origin", "*");
  headers.set("Access-Control-Allow-Methods", "GET, HEAD, OPTIONS");
  headers.set("Access-Control-Allow-Headers", "*");
  headers.set("Access-Control-Max-Age", "86400");
  return new Response(null, { status: 204, headers });
}
```