



โครงการ

เรื่อง

การออกแบบระบบตรวจวัดและควบคุมโรงเรือนโดยใช้ระบบ IOT
(IOT-based Greenhouse Monitoring and Control System Design)

โดย

นายกรดร น้ำสังค์ รหัสนักศึกษา 6415123324

นายกราดร ลุงช้อย รหัสนักศึกษา 6415123325

นายวชรพล คำลัน รหัสนักศึกษา 6415123335

หลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาพลังงานทดแทน

วิทยาลัยพลังงานทดแทน มหาวิทยาลัยแม่โจ้

ปีการศึกษา 2567



โครงการ

เรื่อง

การออกแบบระบบตรวจวัดและควบคุมโรงเรือนโดยใช้ระบบ IOT
(IOT-based Greenhouse Monitoring and Control System Design)

โดย

นายกรดร น้ำสงค์ รหัสนักศึกษา 6415123324

นายกรادر ลุงช้อย รหัสนักศึกษา 6415123325

นายวชรพล คำลัน รหัสนักศึกษา 6415123335

หลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาพลังงานทดแทน

วิทยาลัยพลังงานทดแทน มหาวิทยาลัยแม่โจ้

ปีการศึกษา 2567



การออกแบบระบบตรวจวัดและควบคุมโรงเรือนโดยใช้ระบบ IOT (IOT-based Greenhouse Monitoring and Control System Design)

นายกราด น้ำสังค์ รหัสนักศึกษา 6415123324

นายกราด ลุงช้อย รหัสนักศึกษา 6415123325

นายวัชรพล คำลัน รหัสนักศึกษา 6415123335

โครงการนี้ได้รับการพิจารณาอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษา¹
ตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาพลังงานทดแทน
วิทยาลัยพลังงานทดแทน มหาวิทยาลัยแม่โจ้

คณะกรรมการสอบโครงการ

..... อาจารย์ที่ปรึกษาหลัก
(รองศาสตราจารย์ ดร.ชวโรจน์ ใจสิน)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ธงชัย มณีชูเกตุ)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุลักษณ์ มงคล)

..... คณบดีวิทยาลัย พลังงานทดแทน
(ผู้ช่วยศาสตราจารย์ ดร.นิกราน หอมดวง)

วัน เดือน ปี ที่ทำการสอบ วันสอบ
01/11/2567

③ ลิขสิทธิ์ของวิทยาลัยพลังงานทดแทน
มหาวิทยาลัยแม่โจ้

กิตติกรรมประกาศ

โครงการนี้สำเร็จได้ด้วยความกรุณาจาก รองศาสตราจารย์ ดร.ชวโรจน์ ใจสิน อาจารย์ที่ปรึกษา โครงการที่ค่อยให้คำปรึกษา คำแนะนำ และความช่วยเหลือในทุกด้าน ตลอดจนตรวจสอบโครงการนี้จนสำเร็จ สมบูรณ์ รวมไปถึงบทความทางวิชาการทุกบทความที่นำมาอ้างอิง และนำมารีบกษา ผู้จัดทำ โครงการขอรับ ขอบพระคุณเป็นอย่างยิ่ง ณ โอกาสนี้

ขอขอบพระคุณ ที่กรุณาดำเนินกรรมการการสอบโครงการในครั้งนี้ ทั้งยังให้คำแนะนำที่เป็นประโยชน์ แก่ผู้จัดทำโครงการ สุดท้ายนี้ผู้จัดทำโครงการหวังเป็นอย่างยิ่งว่าโครงการเล่มนี้จะเป็นประโยชน์แก่ผู้ที่สนใจ หากผิดพลาดประการใด ผู้จัดทำโครงการขอภัยไว้ ณ ที่นี่

นายกรดร น้ำสังค์

นายกราดร ลุงช้อย

นายวัชรพล คำลัน

บทคัดย่อ

ชื่อโครงการ	การออกแบบระบบตรวจวัดและควบคุมโรงเรือนโดยใช้ระบบ IoT	
ผู้ทำโครงการ	นายกรดร น้ำสังค์	รหัสนักศึกษา 6415123324
	นายกราดร ลุงช้อย	รหัสนักศึกษา 6415123325
	นายวชรพล คำลัน	รหัสนักศึกษา 6415123335
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.ชวโรจน์ ใจสิน	
กรรมการ	ผู้ช่วยศาสตราจารย์ ดร.ธงชัย มณีชูเกตุ ผู้ช่วยศาสตราจารย์ ดร.สุลักษณา มงคล	

งานวิจัยนี้นำเสนอการออกแบบและพัฒนาระบบตรวจวัดและควบคุมโรงเรือนที่ใช้เทคโนโลยีอินเทอร์เน็ตในทุกสิ่ง เพื่อออกแบบหน้าแสดงผลและวิเคราะห์การใช้พลังงานภายในโรงเรือน โดยใช้ IoT เป็นหัวใจหลักในการควบคุมและตรวจวัดสภาพแวดล้อมในโรงเรือน และมีการแจ้งเตือนแบบเรียลไทม์ผ่านเว็บเบราว์เซอร์หรือมือถือ สามารถปรับค่าอัตโนมัติได้ และเก็บข้อมูลได้ โดยใช้อุปกรณ์ต่างๆ เช่น BME280 ใช้สำหรับวัดอุณหภูมิและความชื้นในโรงเรือน, BH1750 วัดความเข้มของแสงเพื่อปรับแสงสว่าง, PZEM-004T วัดพลังงานไฟฟ้าสำหรับติดตามการใช้พลังงานโดยละเอียด และ RS485 สำหรับการสื่อสารระหว่างอุปกรณ์ เช่นเซอร์ฟเวอร์ที่มีความเสถียรสูง นอกจากนี้ Raspberry Pi ทำหน้าที่เป็นเซิร์ฟเวอร์หลักในการประมวลผลข้อมูล โดยเชื่อมต่อกับ ESP32 ซึ่งเป็นชิปควบคุมเชื่อมโยงเซ็นเซอร์ทั้งหมด และเชื่อมต่อกับซอฟต์แวร์ Home Assistant ผ่านเครือข่ายไร้สาย ทำให้สามารถควบคุมอุปกรณ์ทั้งหมดได้จากระยะไกล ในส่วนของซอฟต์แวร์ ควบคุมการทำงานของ ESP32 ผ่าน ESP Home และเชื่อมต่อและควบคุมอุปกรณ์ผ่านอินเทอร์เน็ตด้วย Tuya โดยการออกแบบหน้าแสดงผลใช้ ReactJS และ Tailwind CSS ส่วน Node.js และ WebSocket API ช่วยให้การเชื่อมต่อและสื่อสารข้อมูลแบบเรียลไทม์เป็นไปอย่างรวดเร็ว MQTT ใช้เป็นโปรโตคอลส่งข้อมูลระหว่าง Home Assistant กับเครือข่ายอื่นภายในระบบ Cloudflare ทำหน้าที่รักษาความปลอดภัยและเพิ่มประสิทธิภาพการเข้าถึงข้อมูลออนไลน์จากภายนอกเครือข่าย เพื่อให้การควบคุมข้อมูลเป็นไปอย่างปลอดภัย ช่วยเพิ่มประสิทธิภาพการวิเคราะห์ข้อมูลเชิงลึกเพื่อนำไปสู่การปรับปรุงในอนาคตและลดการใช้พลังงานอย่างมีประสิทธิภาพ

ผลการทดลองเบื้องต้นในโรงเรือนกัญชาพบว่า ระบบนี้ช่วยรักษาสภาพแวดล้อมที่สม่ำเสมอได้ดียิ่งขึ้น ส่งผลให้คุณภาพของพืชดีขึ้นและลดต้นทุนการดำเนินงาน ระบบบันทึกและแสดงให้เห็นถึงศักยภาพของเทคโนโลยี IoT ในการเพิ่มประสิทธิภาพการเกษตรผ่านการควบคุมสภาพแวดล้อมอย่างแม่นยำ การตรวจสอบแบบเรียลไทม์ และการเพิ่มประสิทธิภาพการใช้พลังงานในโรงเรือน

คำสำคัญ: ระบบตรวจวัด โรงเรือน อินเทอร์เน็ตทุกสรรพสิ่ง

ABSTRACT

Project Title	IOT-based Greenhouse Monitoring and Control System Design	
Author	Mr. Paradon Namsong	Student ID 6415123324
	Mr. Paradorn Lungchoy	Student ID 6415123325
	Mr. Watcharapon Khamlon	Student ID 6415123335
Adviser	Associate Professor Dr. Chawaroj Jaisin	
Committee	Assistant Professor Dr. Thongchai Maneechukate Assistant Professor Dr. Sulaksana Mongkon	

This research presents the design and development of a monitoring and control system for greenhouses utilizing Internet of Things (IoT) technology. The aim is to create a user dashboard and analyzing energy consumption within the greenhouse. The system features real-time alerts via web browsers or mobile devices and allows for automatic adjustments and data collection using various devices, such as the BME280 for measuring temperature and humidity, the BH1750 for measuring light intensity, the PZEM-004T for measuring electrical energy, and RS485 for device communication. A Raspberry Pi acts as the main server for processing, alongside an ESP32 microcontroller chip that connects to all sensors and a wireless network integrated with Home Assistant software for remote user control. ESP Home is used to create software for controlling the ESP32, while Tuya connects and controls devices over the internet. ReactJS and Tailwind CSS design the user interface, and Node.js with WebSocket API facilitates real-time data communication. MQTT serves as the protocol for data transmission between Home Assistant and other networks, and Cloudflare enhances online data access from outside the network. This comprehensive system improves data analysis capabilities for future enhancements and effectively reduces energy consumption.

Preliminary trials in a greenhouse show that this system helps maintain more consistent environmental conditions, leading to improved crop quality and reduced operational costs. This system demonstrates the potential of IoT technology in enhancing agricultural efficiency through precise environmental control, real-time monitoring, and energy optimization in greenhouses.

Keywords: Monitoring system, Greenhouse, IoT

สารบัญ

หน้า

กิตติกรรมประกาศ	ก
บทคัดย่อ	ข
ABSTRACT	ค
สารบัญ.....	ง
สารบัญรูป	ช
สารบัญรูป (ต่อ).....	ช
สารบัญรูป (ต่อ).....	ณ
สารบัญตาราง.....	ญ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของงานวิจัย	1
1.2 วัตถุประสงค์.....	3
1.3 ขอบเขตการศึกษา	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
1.5 ระยะเวลาและสถานที่ดำเนินงาน.....	3
1.6 สถานที่ดำเนินงาน	3
1.7 รายการสัญลักษณ์ย่อ.....	4
บทที่ 2 ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง	5
2.1 ทฤษฎีที่เกี่ยวข้อง.....	5
2.1.1 การคำนวณทางพลังงาน.....	5
2.1.2 ระบบใช้พลังงาน.....	5
2.1.3 เชนเชอร์	7
2.1.4 เชนเชอร์ Tuya	11
2.1.5 Micro controller	12
2.1.6 Software.....	14
2.1.7 Data Communication	21
2.1.8 การดำเนินการเชิงสถิติ	26
2.2 ทบทวนวรรณกรรมที่เกี่ยวข้อง.....	28
บทที่ 3 ขั้นตอนและวิธีดำเนินการ	31
3.1 ขั้นตอนการดำเนินการ.....	31

สารบัญ (ต่อ)

3.1.1 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	32
3.1.2 สำรวจอุปกรณ์และจุดตรวจวัดภายในโรงเรือน	32
3.1.3 จำลองแผนผังการวางแผนอุปกรณ์	34
3.1.4 การออกแบบจุดติดตั้งเซนเซอร์หรือเลย์เอาท์จุดตรวจวัดต่างๆ	35
3.1.5 ติดตั้ง Home assistant server	36
3.1.6 หลักการทำงานของหน้าแสดงผล	44
3.1.7 ออกแบบหน้าแสดงผล.....	45
3.1.8 นำค่าจากเซนเซอร์มาแสดงผล	46
3.1.9 วิเคราะห์และทดสอบประสิทธิภาพ	65
3.1.10 การบันทึกและวิเคราะห์ผล	66
3.1.11 สรุปผลและจัดทำรายงานผลโครงการ	66
3.2 แผนการดำเนินงาน.....	67
บทที่ 4 ผลการทดลอง.....	68
4.1 หน้าแสดงผลข้อมูล	68
4.1.1 Home assistant UI.....	69
4.1.2 Mobile	70
4.1.3 Website	71
4.2 เซนเซอร์ในHome assistant	72
4.3 ข้อมูลสภาพแวดล้อมภายในโรงเรือนกัญชา	73
4.3.1 อุณหภูมิภายในโรงเรือนกัญชา.....	73
4.3.2 ความชื้นสัมพัทธ์ภายในโรงเรือนกัญชา	77
4.4 ข้อมูลสภาพแวดล้อมภายในโรงเรือนบ่อกุ้ง	81
4.4.1 ความส่องสว่างของบ่อกุ้ง.....	81
4.4.2 Dissolved oxygen ของบ่อกุ้ง	82
4.4.3 อุณหภูมน้ำของบ่อกุ้ง	82
4.5 การใช้พลังงาน	83
4.5.1 การใช้พลังงานของโรงเรือนกัญชา.....	83
4.5.2 การใช้พลังงานของโรงเรือนบ่อ กุ้ง	86
4.6 ประสิทธิภาพของระบบ	87

สารบัญ (ต่อ)

บทที่ 5 สรุปผลการทดลอง.....	88
5.1 สรุปผลการทดลอง.....	88
5.2 ข้อเสนอแนะ.....	89
เอกสารอ้างอิง	90
ภาคผนวก ก ข้อมูลผลการวิจัย	91
ภาคผนวก ข ประวัติผู้จัดทำโครงการ.....	93
ประวัติผู้จัดทำโครงการคนที่ 1	94
ประวัติผู้จัดทำโครงการคนที่ 2	95
ประวัติผู้จัดทำโครงการคนที่ 3	96

สารบัญรูป

หน้า

รูปที่ 1.1 รายงานสถิติการจำหน่ายไฟฟ้ารายปีของประเทศไทย	1
รูปที่ 1.2 แผนบูรณาการพัฒนาระยะยาวของประเทศไทย พ.ศ. 2558-2579	2
รูปที่ 2.1 หลักการทำงานของระบบ Evaporative cooling system	6
รูปที่ 2.2 หลอดไฟ LED	7
รูปที่ 2.3 BME280	8
รูปที่ 2.4 FST100-2201 sensor	9
รูปที่ 2.5 VMS-3000-FSJT-N01	10
รูปที่ 2.6 Dissolved Oxygen Sensor	11
รูปที่ 2.7 Tuya Compteur Digital Electric	11
รูปที่ 2.8 Tuya smart switch	12
รูปที่ 2.9 Raspberry PI 4 model B	13
รูปที่ 2.10 Esp32	14
รูปที่ 2.11 หลักการทำงานของ home assistant	15
รูปที่ 2.12 ESP home structure	15
รูปที่ 2.13 Tuya architecture	16
รูปที่ 2.14 การคอมไพล์ CSS ด้วย Tailwind	18
รูปที่ 2.15 การทำงานของ RS485 แบบ Network	23
รูปที่ 3.1 แผนภาพขั้นตอนการดำเนินงาน	31
รูปที่ 3.2 สำรวจอุปกรณ์และจุดตรวจภายในโรงเรือน 1	32
รูปที่ 3.3 MCB Climate controller	32
รูปที่ 3.4 ตู้ระบบแสงสว่าง	33
รูปที่ 3.5 Outdoor Weather Station	33
รูปที่ 3.6 แผนผังระบบแสงสว่างและระบบนำของโรงเรือนกัญชา	34
รูปที่ 3.7 แผนผังจุดติดตั้งเซนเซอร์ของโรงเรือนกัญชา	35
รูปที่ 3.8 แผนผังจุดติดตั้งเซนเซอร์ของบ่อกุ้ง	36
รูปที่ 3.9 หน้าติดตั้ง Raspberry Pi imager	37
รูปที่ 3.10 เลือกบอร์ด Raspberry Pi	38
รูปที่ 3.11 การติดตั้ง Home assistant	38
รูปที่ 3.12 ขั้นตอนการเลือก SD Card	39
รูปที่ 3.13 การเชื่อมต่อของ Raspberry Pi5	40
รูปที่ 3.14 การเชื่อมต่อให้ Raspberry Pi เป็นเซิฟเวอร์หลักของ Home assistant	41

สารบัญรูป (ต่อ)

รูปที่ 3.15 หน้า Login ของ Home assistant	42
รูปที่ 3.16 เพิ่ม cloudflare add on.....	42
รูปที่ 3.17 Token ใน Cloudflare	43
รูปที่ 3.18 Cloud flare tunnel	43
รูปที่ 3.19 หน้าจอแสดงผลพร้อมใช้งาน	44
รูปที่ 3.20 หลักการทำงานของหน้าแสดงผล	45
รูปที่ 3.21 แบบหน้าแสดงผลโรงเรือนกัญชา.....	45
รูปที่ 3.22 แบบหน้าแสดงผลบ่อกุ้ง	46
รูปที่ 3.23 การค้นหาเซนเซอร์ใน Tuya Smart.....	47
รูปที่ 3.24 การเพิ่มเซนเซอร์ใน Tuya Smart	47
รูปที่ 3.25 ขั้นตอนการนำค่า User Code จาก Tuya Smart	49
รูปที่ 3.26 การเพิ่ม Tuya ลงใน Home assistant.....	49
รูปที่ 3.27 การเพิ่ม ESP Home ลงใน Home assistant.....	50
รูปที่ 3.28 หน้าติดตั้ง Nodejs	51
รูปที่ 3.29 หน้าติดตั้ง Visual studio code	52
รูปที่ 3.30 หน้าติดตั้ง Git.....	52
รูปที่ 3.31 หน้าลง Dose3 ใน command prompt	53
รูปที่ 3.32 บุ๊มเลือกเพื่อเปิด Visual studio code.....	54
รูปที่ 3.33 Structure ของ reactjs.....	56
รูปที่ 3.34 หน้าติดตั้ง package	57
รูปที่ 3.35 การขอ Token ใน Home assistant.....	58
รูปที่ 3.36 สื่อสารข้อมูลแบบ Socket	59
รูปที่ 3.37 สร้างตัวแปรเก็บข้อมูล	59
รูปที่ 3.38 นำข้อมูลมาแสดงผล	60
รูปที่ 3.39 Home assistant API swagger document.....	61
รูปที่ 3.40 การดึงข้อมูลจาก Home assistant แบบ Restful API	62
รูปที่ 3.41 รันโค้ดด้วย node package module	63
รูปที่ 3.42 หน้าแสดงผลบนбраузอร์	63
รูปที่ 3.43 เช็ต progressive web app	64
รูปที่ 3.44 การ build project ด้วย Vite	65
รูปที่ 3.45 code สำหรับทดสอบประสิทธิภาพ	65
รูปที่ 3.46 ตัวอย่างผลการทดสอบประสิทธิภาพ.....	66

สารบัญรูป (ต่อ)

รูปที่ 4.1 หน้าแสดงผลข้อมูลของโรงเรือนกัญชา.....	68
รูปที่ 4.2 หน้าแสดงผลของบ่อกุ้ง	69
รูปที่ 4.3 หน้าแสดงผลข้อมูลบน Home assistant	70
รูปที่ 4.4 หน้าแสดงผลบนมือถือ	71
รูปที่ 4.5 หน้าแสดงผลบนบราวเซอร์.....	71
รูปที่ 4.6 Tuya sensor.....	72
รูปที่ 4.7 ESP Home sensor	72
รูปที่ 4.8 กราฟอุณหภูมิภายในโรงเรือน	73
รูปที่ 4.9 อุณหภูมิจุดที่ 1	74
รูปที่ 4.10 อุณหภูมิจุดที่ 2	75
รูปที่ 4.11 อุณหภูมิจุดที่ 3	75
รูปที่ 4.12 อุณหภูมิจุดที่ 4	76
รูปที่ 4.13 อุณหภูมิจุดที่ 5.....	76
รูปที่ 4.14 กราฟความชื้นสัมพัทธ์ในโรงเรือนกัญชา	77
รูปที่ 4.15 ความชื้นสัมพัทธ์จุดที่ 1	78
รูปที่ 4.16 ความชื้นสัมพัทธ์จุดที่ 2.....	79
รูปที่ 4.17 ความชื้นสัมพัทธ์จุดที่ 3.....	80
รูปที่ 4.18 ความชื้นสัมพัทธ์จุดที่ 4.....	80
รูปที่ 4.19 ความชื้นสัมพัทธ์จุดที่ 5	81
รูปที่ 4.20 ความส่องสว่างโรงเรือนบ่อ กุ้ง	82
รูปที่ 4.21 Dissolved oxygen โรงเรือนบ่อ กุ้ง	82
รูปที่ 4.22 อุณหภูมน้ำของบ่อ กุ้ง	83
รูปที่ 4.23 กำลังไฟฟ้าของปั๊ม.....	83
รูปที่ 4.24 กำลังไฟฟ้าของระบบไฟฟ้าแสงสว่าง	84
รูปที่ 4.25 พลังงานไฟฟ้าของโรงเรือน	84
รูปที่ 4.26 การใช้พลังงานรวมของบ่อ กุ้ง	87

สารบัญตาราง

หน้า

ตารางที่ 3.1 แผนการดำเนินงาน.....	67
ตารางที่ 4.1 ข้อมูลอุณหภูมิในแต่ละจุดของโรงเรือน	73
ตารางที่ 4.2 ข้อมูลความชื้นสัมพัทธ์ภายในโรงเรือน	78
ตารางที่ 4.3 ข้อมูลการใช้พลังงานของโรงเรือนกัญชา.....	85
ตารางที่ 4.4 ผลผลิตตลอด 3 เดือน.....	85
ตารางที่ 4.5 ผลการทดสอบประสิทธิภาพ	87

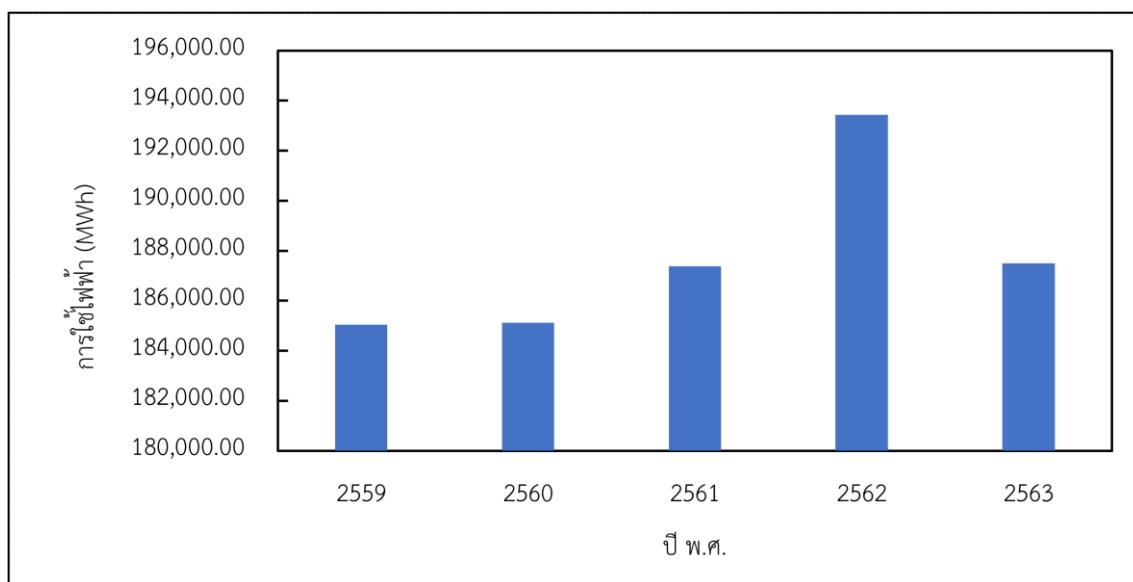
บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของงานวิจัย

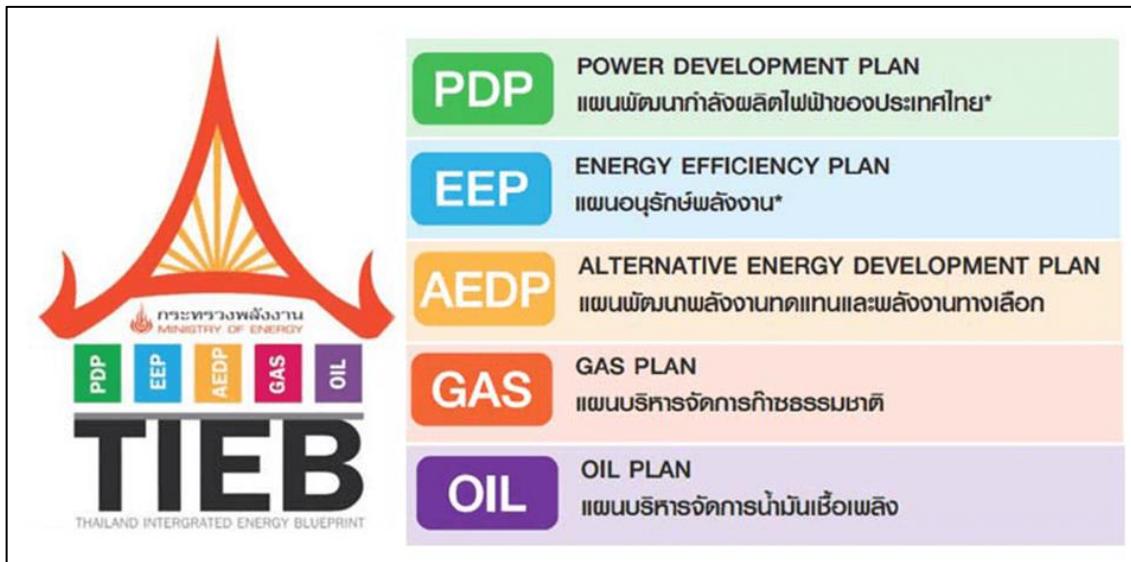
การตรวจสอบสภาพแวดล้อมภายในโรงเรือน เช่น อุณหภูมิ ความชื้น และสภาพแสง เป็นปัจจัยสำคัญที่มีผลต่อการเจริญเติบโตของพืชและสัตว์ อย่างไรก็ตาม การตรวจสอบแบบดั้งเดิมมักจะต้องใช้แรงงานคน มีความเสี่ยงต่อข้อผิดพลาด และไม่สามารถตรวจสอบได้ตลอดเวลา ทำให้ขาดข้อมูลที่จำเป็นต่อการดูแลรักษาระบบเรือนอย่างมีประสิทธิภาพ การนำเทคโนโลยี Internet of Things (IoT) เข้ามาใช้ช่วยให้สามารถตรวจสอบและเก็บข้อมูลได้อย่างแม่นยำและต่อเนื่อง ช่วยลดข้อผิดพลาดจากมนุษย์ และประหยัดเวลาและค่าใช้จ่ายในการตรวจสอบ

การใช้พลังงานในโรงเรือนเป็นปัญหาสำคัญสำหรับภาคการเกษตรทั่วโลก โรงเรือนต้องใช้พลังงานจำนวนมากเพื่อควบคุมสภาพอากาศภายในโรงเรือนให้เหมาะสมต่อการเจริญเติบโตของพืชและสัตว์เลี้ยง ปัญหาหลักของการใช้พลังงานในโรงเรือน ได้แก่ โรงเรือนต้องใช้พลังงานไฟฟ้าจำนวนมากเพื่อควบคุมอุณหภูมิ ความชื้น แสงสว่าง และการระบายอากาศภายในโรงเรือน ส่งผลให้ต้นทุนการผลิตสูงขึ้น ปัญหาสิ่งแวดล้อม โรงเรือนที่ใช้พลังงานไฟฟ้าจำนวนมากจะปล่อยก๊าซเรือนกระจกสู่ชั้นบรรยากาศ ส่งผลให้เกิดภาวะโลกร้อน ความสิ้นเปลืองพลังงาน โรงเรือนบางแห่งใช้พลังงานอย่างไม่มีประสิทธิภาพ ส่งผลให้เกิดการสิ้นเปลืองพลังงาน ดังแสดงใน รูปที่ 1.1 รายงานสถิติการจำหน่ายไฟฟ้ารายปีของประเทศไทย



รูปที่ 1.1 รายงานสถิติการจำหน่ายไฟฟ้ารายปีของประเทศไทย

จากปัญหาที่กล่าวมาแล้วข้างต้นจึงเป็นที่มาของงานวิจัยนี้ที่ต้องการออกแบบระบบตรวจวัดและควบคุมโรงเรือนโดยใช้ระบบ IoT ซึ่งสอดคล้องกับแผนอนุรักษ์พลังงาน (EEP) ดังแสดงในรูปที่ 1.2 แผนบูรณาการพลังงานระยะยาวของประเทศไทย พ.ศ. 2558-2579



รูปที่ 1.2 แผนบูรณาการพลังงานระยะยาวของประเทศไทย พ.ศ. 2558-2579

(กระทรวงพลังงาน, 2563)

จากปัญหาที่กล่าวมาแล้วข้างต้นจึงเป็นที่มาของงานวิจัยนี้ที่ต้องการ ระบบตรวจวัดและจัดการพลังงานที่ใช้ในโรงเรือนพลังงานแสงอาทิตย์ พลังงานลม และพลังงานชีวภาพ ช่วยลดการพึ่งพาพลังงานไฟฟ้าจากแหล่งเชื้อเพลิง fosซิล การใช้เทคโนโลยีประยุกต์พลังงาน การใช้เทคโนโลยีประยุกต์พลังงาน เช่น มอเตอร์ประสิทธิภาพสูง โคมไฟ LED และระบบควบคุมอัตโนมัติ ช่วยลดการใช้พลังงานลง การจัดการพลังงานอย่างมีประสิทธิภาพ การจัดการพลังงานอย่างมีประสิทธิภาพ เช่น การวางแผนการใช้พลังงาน การบำรุงรักษาอุปกรณ์ และการฝึกอบรมพนักงาน ช่วยลดการสิ้นเปลืองพลังงาน เป็นแนวทางการแก้ไขปัญหาดังกล่าว

การเปลี่ยนไปใช้ IoT ใน การตรวจสอบภายในโรงเรือนมีความสำคัญอย่างยิ่ง เนื่องจากช่วยให้สามารถเฝ้าระวังสภาพแวดล้อมได้แบบเรียลไทม์ ลดข้อผิดพลาด เพิ่มความรวดเร็วในการตอบสนองเมื่อสภาพแวดล้อมไม่เหมาะสม และสามารถเก็บข้อมูลระยะยาวเพื่อวิเคราะห์และปรับปรุงกระบวนการได้ การเปลี่ยนแปลงนี้ไม่เพียงแต่ช่วยลดต้นทุนและประหยัดเวลา แต่ยังส่งเสริมการใช้ทรัพยากรอย่างคุ้มค่า และเพิ่มประสิทธิภาพการจัดการโรงเรือนให้ได้ผลผลิตที่ดี

1.2 วัตถุประสงค์

- เพื่อออกแบบและพัฒนาระบบแสดงผลและวิเคราะห์ผลสภาพแวดล้อมและการใช้พลังงานสำหรับโรงเรือนปลูกพืชและเลี้ยงสัตว์น้ำ

1.3 ขอบเขตการศึกษา

- ระบบตรวจวัดใช้สำหรับแสดงผลตรวจวัดสภาพแวดล้อมและการใช้พลังงานสำหรับโรงเรือนปลูกพืชและเลี้ยงสัตว์น้ำ
- ระบบตรวจวัดการใช้พลังงานไฟฟ้าได้แก่ ระบบ EVAP, ระบบแสงสว่าง, ระบบจ่ายน้ำ-ปุ่ยและพัดลมระบบอากาศ
- ระบบตรวจวัดสภาพแวดล้อมประกอบด้วยอุณหภูมิเวดล้อมภายในและภายนอกโรงเรือน, ความชื้นและภายนอกโรงเรือน, อุณหภูมิภายในน้ำ และอุณหภูมิในน้ำ
- ระบบเซิร์ฟเวอร์หลักใช้ Raspberry Pi ติดตั้งระบบปฏิบัติการเป็น Home assistant จุดตรวจวัดย่อยใช้อุปกรณ์ชุดเซนเซอร์ของ TUYA และ ESPHome
- ออกแบบและพัฒนาระบบแสดงผลและวิเคราะห์ผลสภาพแวดล้อมสำหรับโรงเรือนปลูกพืชและเลี้ยงสัตว์น้ำ
- พัฒนาระบบแสดงผลและวิเคราะห์ผลสภาพแวดล้อมสำหรับโรงเรือนปลูกพืชและเลี้ยงสัตว์น้ำ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- สามารถตรวจวัดการใช้พลังงานในโรงเรือนในแต่ละส่วนและวิเคราะห์ผลได้

1.5 ระยะเวลาและสถานที่ดำเนินงาน

ระยะเวลาในการดำเนินงาน 5 เดือน

ตั้งแต่วันที่ 22 เดือน พฤศจิกายน พ.ศ.2566

ถึงวันที่ 30 เดือน เมษายน พ.ศ.2567

1.6 สถานที่ดำเนินงาน

- วิทยาลัยพลังงานทดแทน มหาวิทยาลัยแม่โจ้
- พาร์ม 900 ไร่ มหาวิทยาลัยแม่โจ้

1.7 รายการสัญลักษณ์ย่อ

สัญลักษณ์	ความหมาย	หน่วย
\bar{X}	ค่าเฉลี่ย	-
μ	ค่าเฉลี่ยของประชากร	-
n	จำนวนค่าทั้งหมดในชุดข้อมูล	-
N	จำนวนค่าทั้งหมดในประชากร	-
P	กำลังไฟฟ้า	W
I	กระแสไฟฟ้า	A
V	แรงดันไฟฟ้า	V
P.F.	ตัวประกอบกำลังไฟฟ้า	-

บทที่ 2

ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 การคำนวณทางพลังงาน

ใช้ในการคำนวณกำลังไฟฟ้าในระบบไฟฟ้ากระแสสลับ (AC) โดยมีความสัมพันธ์กับแรงดันไฟฟ้า กำลังไฟฟ้า (หน่วยเป็นวัตต์, W) คือ พลังงานที่ใช้งานในระบบไฟฟ้าต่อหนึ่งหน่วยเวลา ในระบบไฟฟ้ากระแสตรง (DC) สูตรในการคำนวณกำลังไฟฟ้าคือ แรงดันไฟฟ้า (หน่วยเป็นโวลต์, V) เป็นพลังงานที่จำเป็นต่อการเคลื่อนย้ายประจุไฟฟ้าผ่านวงจร หรือสามารถเข้าใจว่าเป็น "ความดัน" ที่ผลักให้กระแสไฟฟ้าไหลในวงจรกระแสไฟฟ้า (หน่วยเป็นแอมป์เรีย, A) คือ อัตราการไหลของประจุไฟฟ้าผ่านจุดใดจุดหนึ่งในวงจรไฟฟ้าตัวตัวประกอบกำลังเป็นค่าที่แสดงถึงประสิทธิภาพของการใช้พลังงานในระบบไฟฟ้ากระแสสลับ (AC) มีค่าอยู่ระหว่าง 0 ถึง 1 ค่านี้แสดงสัดส่วนระหว่างกำลังไฟฟ้าที่ใช้งานจริง (Real Power) กับ กำลังไฟฟาร่วมที่จ่ายให้ระบบ (Apparent Power) ถ้า P.F. มีค่าใกล้เคียงกับ 1 แสดงว่าระบบใช้พลังงานอย่างมีประสิทธิภาพสูงสุดทฤษฎีเบื้องหลัง ในระบบไฟฟ้ากระแสสลับ (AC) จะมีทั้งกำลังไฟฟ้าที่ใช้งานได้จริง (Real Power) และกำลังไฟฟ้าเชิงปฏิกริยา (Reactive Power) ซึ่งเกิดจากการทำงานของอุปกรณ์ที่มีความต้านทานเชิงปฏิกริยา เช่น คาปาซิเตอร์และอินดักเตอร์ ส่งผลให้เกิดเฟสต่างกันระหว่างกระแสไฟฟ้าและแรงดันไฟฟ้าการคำนวณกำลังไฟฟ้าใน AC จึงต้องคำนึงถึงตัวประกอบกำลัง จึงเป็นการคำนวณกำลังไฟฟ้าที่คำนึงถึงการสูญเสียพลังงานในวงจรเชิงปฏิกริยา โดยมีการคูณตัวประกอบกำลัง (ซึ่งจะทำให้กำลังไฟฟ้าที่คำนวณได้ตามสัดส่วนที่เกิดจากการสูญเสียพลังงานเชิงปฏิกริยา) ดังสมการที่ 2.1

$$P=IV \times P.F.$$

สมการที่ 2.1

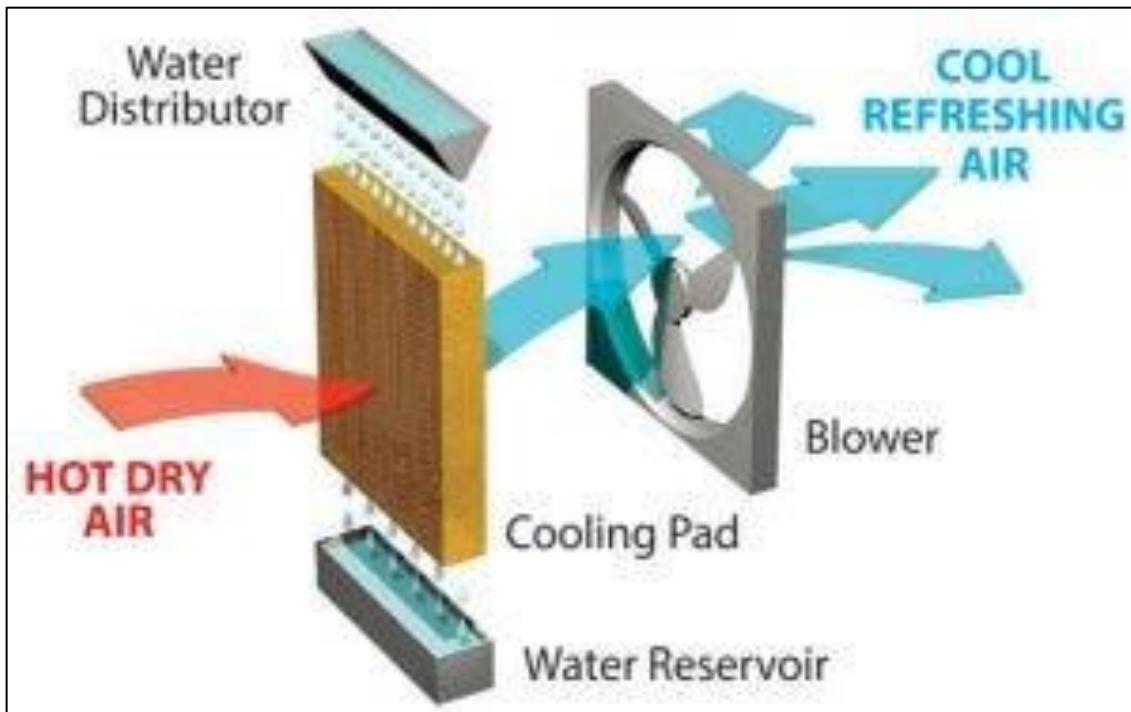
เมื่อ	P	คือ กำลังไฟฟ้า (W)
	I	คือ กระแสไฟฟ้า (A)
	V	คือ แรงดันไฟฟ้า (V)
	P.F.	คือ ตัวประกอบกำลังไฟฟ้า

2.1.2 ระบบใช้พลังงาน

2.1.2.1 ระบบปั๊มทำความเย็น

ระบบ Evaporative cooling system หลักการทำงานของระบบ อีแวนป์ คือ การปล่อยกระแสลมให้ผ่านตัวกลางที่มีน้ำให้ผ่าน คูลลิ่งแพด (Cooling pad) และการที่อากาศซึ่งมีอุณหภูมิสูงกว่าให้ผ่านน้ำจะทำให้น้ำระเหยกลายเป็นไออกซ์เจน ซึ่งจะเป็นการดึงความร้อนของอากาศออกทำให้อุณหภูมิของอากาศ

ลดลงจากหลักการนี้ทำให้มีการนำมารือกแบบใช้กับการทำโรงเรือนเลี้ยงสัตว์หรือปลูกพืชได้ โดยมีการติดตั้งแผงความเย็น คลูลิ่งแพด (Cooling pad) ที่ทำจากกระดาษอัดเคลือบน้ำยาชนิดพิเศษ ซึ่งกระดาษอัดชนิดนี้ได้มีการนำมาเรียงเป็นแผ่นมีความหนาประมาณ 4-6 มม. และวางเรียงให้มีพื้นที่ผิวมากที่สุด เพื่อให้มีโอกาสสัมผัสน้ำและอากาศที่ไหลผ่านมากที่สุด จะเป็นการช่วยให้น้ำมีการระเหยได้มากนั่นเอง และมีพัดลมดูดเอาอากาศร้อนข้างนอกผ่านแผง คลูลิ่งแพด (Cooling pad) ที่มีน้ำไหลผ่าน



รูปที่ 2.1 หลักการทำงานของระบบ Evaporative cooling system

2.1.2.2 ระบบแสงสว่าง

การสร้างระบบแสงสว่างในโรงเรือนปลูกมะเขือเทศเป็นหนึ่งในการใช้เทคโนโลยีที่มีประสิทธิภาพเพื่อเพิ่มผลผลิตและคุณภาพของผลผลิต โดยเฉพาะในสภาพที่มีการเพาะปลูกในสถานที่ปิดกันหรือโรงเรือน ระบบแสงสว่างสามารถช่วยให้พืชได้รับแสงที่เหมาะสมตลอดวันและปีที่เหมาะสมสำหรับการเจริญเติบโตและผลผลิตอย่างมีคุณภาพ นอกจากนี้ยังช่วยลดการใช้พลังงานโดยการใช้แสงเทียมเท่าที่เหมาะสมตามความต้องการของพืช รวมถึงสามารถควบคุมได้ตลอดเวลา เช่น ปรับแสงในเวลากลางคืนเพื่อสร้างสภาพใกล้เคียงกับแสงธรรมชาติเพื่อส่งเสริมการเจริญเติบโตของพืชได้ดีขึ้น

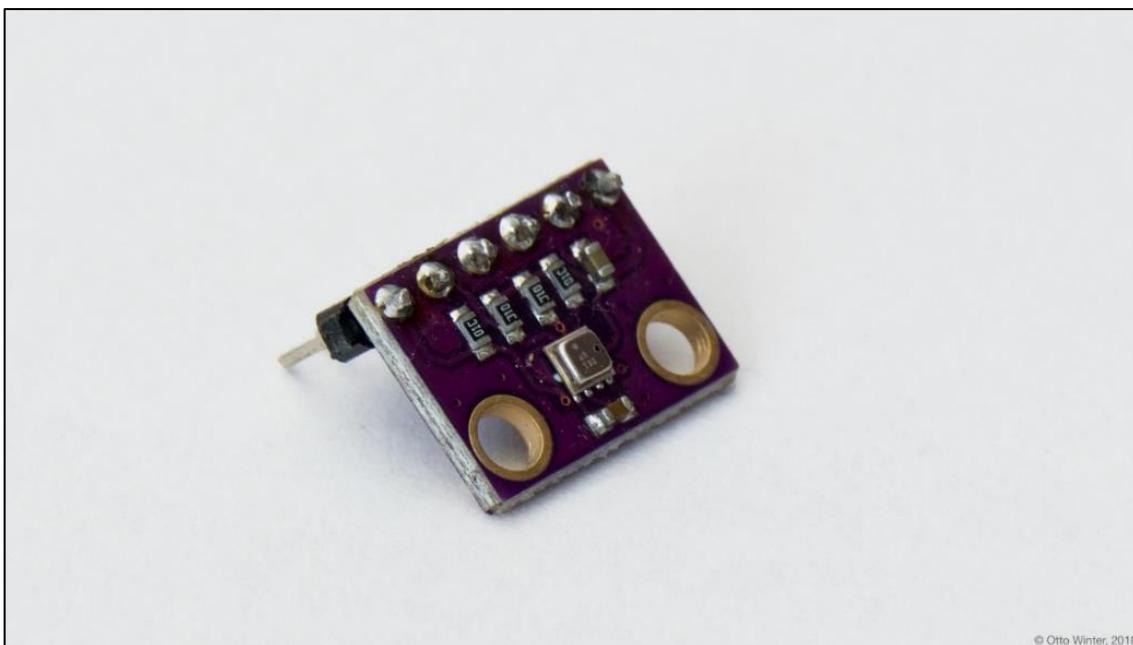


รูปที่ 2.2 หลอดไฟ LED

2.1.3 เซนเซอร์

2.1.3.1 เซนเซอร์วัดอุณหภูมิและความชื้น

เซนเซอร์วัดอุณหภูมิและความชื้น (Temperature and Humidity Sensor) เป็นอุปกรณ์ที่ใช้ในการวัดอุณหภูมิและความชื้นในสภาพแวดล้อม โดยทั่วไปมักใช้ในการตรวจสอบสภาพอากาศภายในบ้าน หรือสภาพแวดล้อมที่ต้องการการควบคุมอุณหภูมิและความชื้น เช่น ในโรงเรือนเพาะปลูกพืช, ห้องปฏิบัติการทางการแพทย์, ห้องอาหาร, หรือโรงงานที่ต้องการควบคุมสภาพแวดล้อม เซนเซอร์ชนิดนี้มีความสำคัญในการทำงานของหลายโครงงาน IoT (Internet of Things) หรือโปรเจคที่เกี่ยวข้องกับการตรวจวัดและความคุมสภาพแวดล้อมต่าง ๆ ที่มีความสำคัญต่อผู้ใช้งานในทุกวงการและสถานการณ์ต่าง ๆ ที่ต้องการการเฝ้าระวังและความคุมสภาพอากาศและสภาพแวดล้อมในการดำเนินกิจกรรมต่าง ๆ อย่างมีประสิทธิภาพและถูกต้อง



© Otto Winter, 2018

รูปที่ 2.3 BME280

2.1.3.2 เชนเซอร์วัดค่าความส่วนของแสง

ชนเซอร์แสงเป็นอุปกรณ์พาร์ซีฟที่เปลี่ยนพลังงานแสงเป็นเอกสาร์พุตสัญญาณไฟฟ้า ชนเซอร์แสงรู้จักในชื่ออุปกรณ์โฟโตอิเล็กทริกหรือโฟโตเซนเซอร์เนื่องจากการเปลี่ยนพลังงานแสง (โฟตอน) เป็นสัญญาณไฟฟ้า (อิเล็กตรอน) โฟโตทรานซิสเตอร์ โฟโตเรซิสเตอร์ และโฟโตไดโอดคือชนเซอร์วัดความเข้มแสงที่ว่าไป โฟโตอิเล็กทริกชนเซอร์จะใช้ลำแสงเพื่อตรวจจับการมีอยู่ของวัตถุ โดยจะเปล่งแสง (ชนิดมองเห็นได้หรืออินฟราเรด) จากส่วนเปล่งแสง โฟโตอิเล็กทริกชนเซอร์ชนิดสะท้อนแสงจะถูกใช้เพื่อตรวจจับลำแสงที่สะท้อนจากชิ้นงาน ลำแสงจะถูกเปล่งออกมากจากส่วนเปล่งแสงจะรับโดยส่วนรับแสง ทั้งสองส่วนนี้จะอยู่ในตัวเรือนเดียวกัน ชนเซอร์จะได้รับแสงที่สะท้อนจากชิ้นงานส่วนโฟโตทรานซิสเตอร์จะใช้ระดับของแสงที่ตรวจจับได้เพื่อตรวจสอบว่ามีกระแสไฟไหลผ่านวงจรเท่าใด หากชนเซอร์อยู่ในห้องมีด จะมีกระแสไฟไหลผ่านเพียงเล็กน้อยเท่านั้น หากตรวจจับแสงสว่างได้ กระแสไฟไหลผ่านจำนวนมากขึ้น โฟโตเรซิสเตอร์ถูกสร้างขึ้นจากแคนเดกเมียมแซลไฟด์ ซึ่งมีความต้านทานสูงสุดเมื่อชนเซอร์อยู่ในที่มีด เมื่อโฟโตเรซิสเตอร์ได้รับแสง ความต้านทานจะลดลงตามความเข้มแสง เมื่อเชื่อมต่อกับวงจรและปรับสมดุลด้วยโพเทนชิโอมิเตอร์ ความเปลี่ยนแปลงของความเข้มแสงจะแสดงเป็นแรงดันไฟฟ้าที่เปลี่ยนแปลง ชนเซอร์นี้มีความเรียบง่าย น่าเชื่อถือ และราคาถูก รวมทั้งมีการใช้งานอย่างกว้างขวางเพื่อใช้วัดความเข้มแสง



รูปที่ 2.4 FST100-2201 sensor

2.1.3.3 เชนเซอร์วัดความเร็วลม

VMS-3000-FSJT-N01 เป็นชนเซอร์ประเภทเอนิมومิเตอร์ (Anemometer) ที่ออกแบบมาสำหรับการวัดความเร็วลม ชนเซอร์ประเภทนี้เหมาะสมสำหรับการใช้งานในงานอุตสาหกรรม, การติดตั้งบนกังหันลม, สถานีตรวจน้ำอากาศ และแอปพลิเคชัน IoT ที่ต้องการการตรวจวัดความเร็วลมที่แม่นยำและต่อเนื่อง คุณสมบัติและโครงสร้างมี ถัว�หมุนสามใบ: VMS-3000-FSJT-N01 มีถัว�หมุนสามใบที่มีน้ำหนักเบาและออกแบบมาให้หมุนได้อย่างอิสระเมื่อมีลมพัดผ่าน โครงสร้างนี้ช่วยให้เกิดความไวต่อความเร็วลมและทำให้ชนเซอร์สามารถจับความเร็วลมได้หลากหลายช่วง ตัวแปลงสัญญาณ (Signal Conversion): เมื่อถัว�หมุน สัญญาณการหมุนนี้จะถูกแปลงเป็นสัญญาณไฟฟ้าหรือสัญญาณพลังซึ่งแปลงสัญญาณเหล่านี้เป็นความเร็วลมที่สามารถนำไปวิเคราะห์ต่อได้ วัสดุกันสนิมและทนทานต่อสภาพแวดล้อม: ชนเซอร์รุ่นนี้ทำจากวัสดุที่ทนทานต่อสภาพอากาศ เช่น พลาสติก ABS หรือโลหะเคลือบป้องกันสนิม จึงเหมาะสมสำหรับการใช้งานภายนอกในสภาพอากาศที่หลากหลาย



รูปที่ 2.5 VMS-3000-FSJT-N01

2.1.3.4 เชนเซอร์วัดค่าออกซิเจนในน้ำ

ชนเซอร์วัดออกซิเจนละลายน้ำ หรือ DO Sensor (Dissolved Oxygen Sensor) เป็นอุปกรณ์ที่ใช้วัดปริมาณออกซิเจนที่ละลายน้ำอยู่ในน้ำ ซึ่งมีความสำคัญต่อการประเมินคุณภาพน้ำในหลายๆ ด้าน เช่น การตรวจสอบสภาพแวดล้อมทางน้ำ การเพาะเลี้ยงสัตว์น้ำ การบำบัดน้ำเสีย และการตรวจสอบระบบนิเวศในแหล่งน้ำธรรมชาติ โดย DO Sensor สามารถออกถึงปริมาณออกซิเจนที่สิ่งมีชีวิตในน้ำสามารถนำไปใช้ในกระบวนการหายใจได้ การวัด DO สามารถทำได้โดยใช้หลักการทำงานเคมีไฟฟ้าหรือทางแสง หลักการของการเกิดปฏิกิริยาเคมีไฟฟ้า ของชนเซอร์ประเภทนี้จะใช้ขั้วไฟฟ้า (Electrodes) ที่ทำจากวัสดุต่างกันสองชนิด เพื่อให้เกิดกระแสไฟฟ้าเมื่อมีออกซิเจนละลายน้ำเข้ามาทำปฏิกิริยาที่ขั้วไฟฟ้า ซึ่งทำให้กระแสไฟฟ้าเปลี่ยนแปลงตามปริมาณออกซิเจนที่ละลายน้ำอยู่ในน้ำ



รูปที่ 2.6 Dissolved Oxygen Sensor

2.1.4 เซนเซอร์ Tuya

2.1.4.1 Tuya Compteur Digital Electric

เป็นอุปกรณ์มิเตอร์ไฟฟ้าดิจิทัลที่เชื่อมต่อกับระบบ Tuya Smart ซึ่งเป็นแพลตฟอร์ม IoT ที่ช่วยในการควบคุมและจัดการอุปกรณ์スマาร์ทโฮมผ่านแอปพลิเคชันหรือระบบควบคุมระยะไกล โดย Compteur Digital Electric หรือ "มิเตอร์ไฟฟ้าดิจิทัล" นี้ได้รับการออกแบบมาเพื่อวัดและบันทึกข้อมูลการใช้พลังงานไฟฟ้าภายในบ้านหรือสถานที่ต่างๆ แบบเรียลไทม์ ช่วยให้ผู้ใช้ทราบถึงการใช้พลังงานและสามารถจัดการพลังงานได้อย่างมีประสิทธิภาพมากขึ้น



รูปที่ 2.7 Tuya Compteur Digital Electric

2.1.4.2 Tuya smart switch

เป็นสวิตช์ไฟอัจฉริยะที่พัฒนาโดย Tuya Smart ซึ่งเป็นแพลตฟอร์ม IoT ที่ช่วยให้ผู้ใช้งานสามารถควบคุมและจัดการอุปกรณ์ไฟฟ้าต่างๆ ในบ้านได้ผ่านสมาร์ทโฟนหรืออุปกรณ์ที่เชื่อมต่อกับอินเทอร์เน็ต เช่น สวิตช์ไฟ พัดลม หรือเครื่องใช้ไฟฟ้าต่างๆ สวิตช์นี้ชื่อมต่อกับเครือข่าย Wi-Fi และใช้งานผ่านแอป Tuya Smart หรือ Smart Life ทำให้ผู้ใช้สามารถควบคุมและตั้งค่าการเปิด-ปิดไฟฟ้าได้จากทุกที่



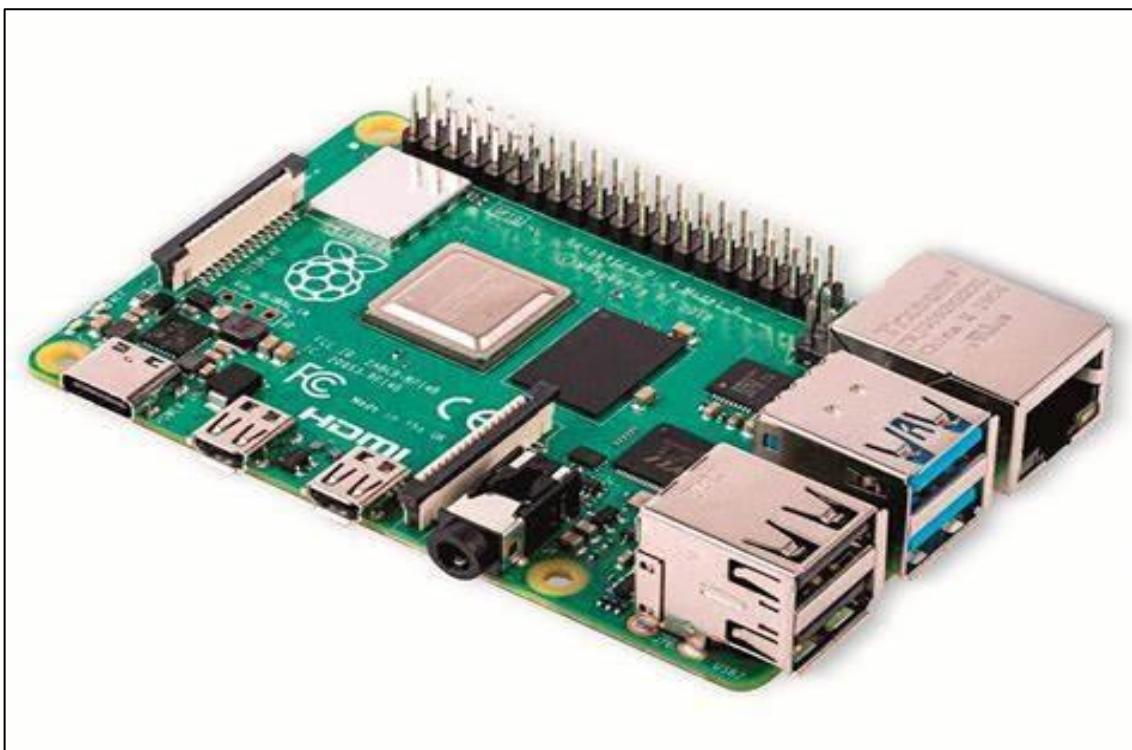
รูปที่ 2.8 Tuya smart switch

2.1.5 Micro controller

2.1.5.1 Raspberry Pi

Raspberry Pi เป็นเครื่องคอมพิวเตอร์ขนาดเล็กที่มีราคาไม่แพงดังแสดงในรูปที่ 2.9 Raspberry Pi 4 model B ที่เหมาะสมสำหรับใช้ในการพัฒนาโครงการต่าง ๆ ด้วยความสามารถในการเชื่อมต่อกับอุปกรณ์ภายนอกและการทำงานกับภาษาโปรแกรมต่าง ๆ ที่หลากหลาย เช่น Python, C/C++, และอื่น ๆ

ที่ช่วยให้ผู้ใช้งานสามารถสร้างโปรเจกต์หลากหลายได้อย่างง่ายดาย มีชุดความรู้และเอกสารการใช้งานที่นิยมอยู่มากมายที่ช่วยให้การเริ่มต้นกับ Raspberry Pi เป็นเรื่องที่ไม่ยากเย็น นอกจาคนี้ยังมีชุมชนผู้ใช้งานที่ใหญ่ในการสนับสนุนและแบ่งปันความรู้ในการใช้งาน Raspberry Pi ทำให้เป็นที่นิยมในการใช้ในโครงการทั้งในส่วนของการศึกษาและในอุตสาหกรรม ในงานโครงงานระบบตรวจวัดและควบคุมโรงเรือนที่ใช้เทคโนโลยี IoT Raspberry Pi ทำหน้าที่เป็นเซิร์ฟเวอร์หลัก สำหรับการประมวลผลข้อมูลที่ได้รับจากเซ็นเซอร์ต่าง ๆ เช่น เซ็นเซอร์วัดอุณหภูมิและความชื้น (BME280), เซ็นเซอร์วัดแสง (BH1750), และเซ็นเซอร์วัดพลังงานไฟฟ้า (PZEM-004T) นอกจากนี้ Raspberry Pi ยังเชื่อมต่อกับชิป ESP32 ซึ่งเป็นชิปที่ควบคุมและเชื่อมโยงเซ็นเซอร์เหล่านี้ผ่านระบบเครือข่ายไร้สาย เพื่อการควบคุมอุปกรณ์ทั้งหมดได้จากระยะไกลผ่านซอฟต์แวร์ Home Assistant ที่ติดตั้งอยู่บน Raspberry Pi



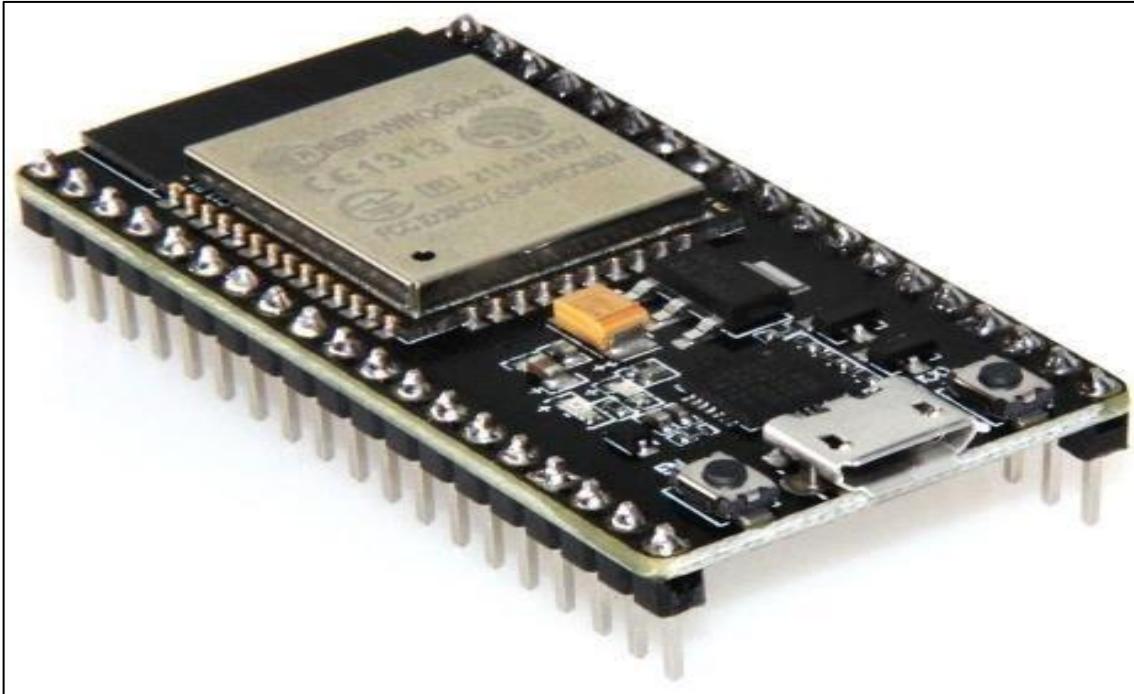
รูปที่ 2.9 Raspberry PI 4 model B

2.1.5.2 ESP32

ESP32 คือชิปเซ็ต (System-on-Chip) ที่พัฒนาโดย Espressif Systems ซึ่งเป็นบริษัทที่ผลิตไมโครคอนโทรลเลอร์และโมดูลเครือข่ายไร้สาย ชิปเซ็ตชุดนี้มีความสามารถหลากหลายในการเชื่อมต่อและการประมวลผล รวมถึงมีความสามารถในการเชื่อมต่อกับเซ็นเซอร์ต่าง ๆ และเครือข่ายไร้สายเช่น Wi-Fi และ Bluetooth อีกด้วย

ESP32 ดังแสดงในรูปที่ 2.10 Esp32 มีประสิทธิภาพสูงและมีความสามารถในการประมวลผลที่ดี ทำให้เหมาะสมสำหรับการใช้งานในโปรเจกต์ที่ต้องการความยืดหยุ่นและประสิทธิภาพ เช่น โครงงาน

อิเล็กทรอนิกส์ที่มีการเชื่อมต่อกับอินเทอร์เน็ต การพัฒนา IoT (Internet of Things), หุ่นยนต์พกพา และอื่น ๆ อีกมากมาย นับเป็นหนึ่งในชิปเซ็ตที่ได้รับความนิยมในการนักพัฒนาและผู้สร้างโปรเจคต่าง ๆ

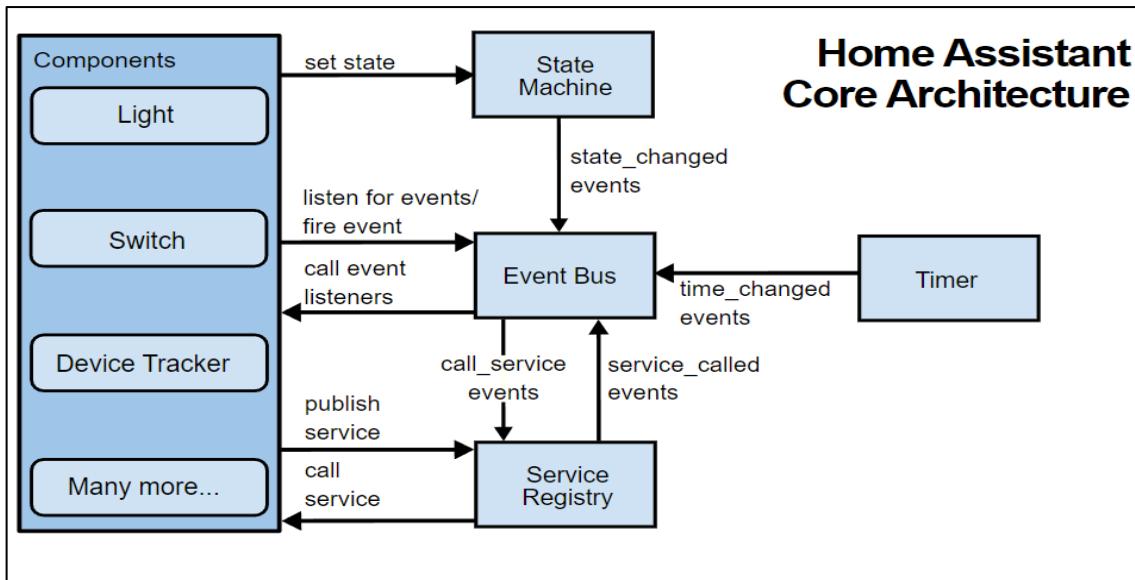


รูปที่ 2.10 Esp32

2.1.6 Software

2.1.6.1 Home assistant core

Home Assistant Core เป็นแพลตฟอร์มซอฟต์แวร์โอเพนซอร์สที่ใช้ในการสร้างระบบอัตโนมัติ ดังแสดงในรูปที่ 2.11 หลักการทำงานของ home assistant สำหรับบ้านหรือสถานที่อื่น ๆ โดยมุ่งเน้นไปที่การควบคุมและจัดการอุปกรณ์อัจฉริยะต่าง ๆ อย่างเช่น หลอดไฟสมาร์ท, เครื่องปรับอากาศ, กล้องวงจรปิด, ประตูหรือหน้าต่างอัจฉริยะ, เครื่องควบคุมอุณหภูมิ, และอุปกรณ์อื่น ๆ ซึ่งสามารถสื่อสารผ่านเครือข่ายได้ เช่น Wi-Fi, Zigbee, Z-Wave, Bluetooth, และ MQTT หลักการทำงานของ Home Assistant Core คือการรวมอุปกรณ์อัจฉริยะต่าง ๆ ให้ทำงานร่วมกันในระบบเดียวกัน โดยใช้โปรแกรม Home Assistant Core เป็นส่วนกลางในการควบคุมและจัดการอุปกรณ์เหล่านั้น ผู้ใช้สามารถกำหนดค่าและสร้างการทำงานอัตโนมัติต่าง ๆ เพื่อให้อุปกรณ์ทำงานตามต้องการ เช่น การเปิด-ปิดอุปกรณ์ตามตารางเวลา, การเปิด-ปิดอุปกรณ์โดยอิสระ

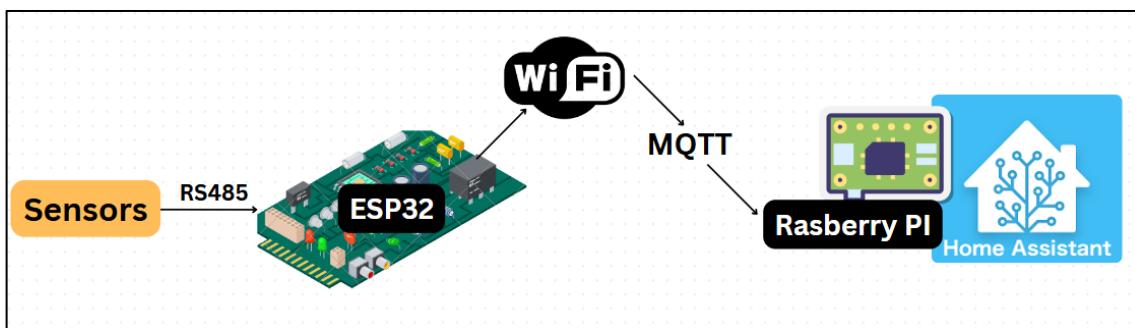


รูปที่ 2.11 หลักการทำงานของ home assistant

2.1.6.2 ESP Home

ESP Home เป็นแพลตฟอร์มโอเพนซอร์สที่ใช้ในการสร้างซอฟต์แวร์สำหรับควบคุมอุปกรณ์อิเล็กทรอนิกส์ด้วยไมโครคอนโทรลเลอร์ ESP8266 และ ESP32 ซึ่งเป็นชิปเซ็ตที่มีความสามารถในการเชื่อมต่อกับเครือข่ายไร้สายได้ โดย ESP Home ช่วยให้ผู้ใช้สามารถสร้างโปรแกรมควบคุมอุปกรณ์ต่าง ๆ ได้อย่างง่ายดายผ่านแพลตฟอร์ม Home Assistant ซึ่งเป็นระบบที่ใช้ในการควบคุมอุปกรณ์ในบ้านอัตโนมัติ

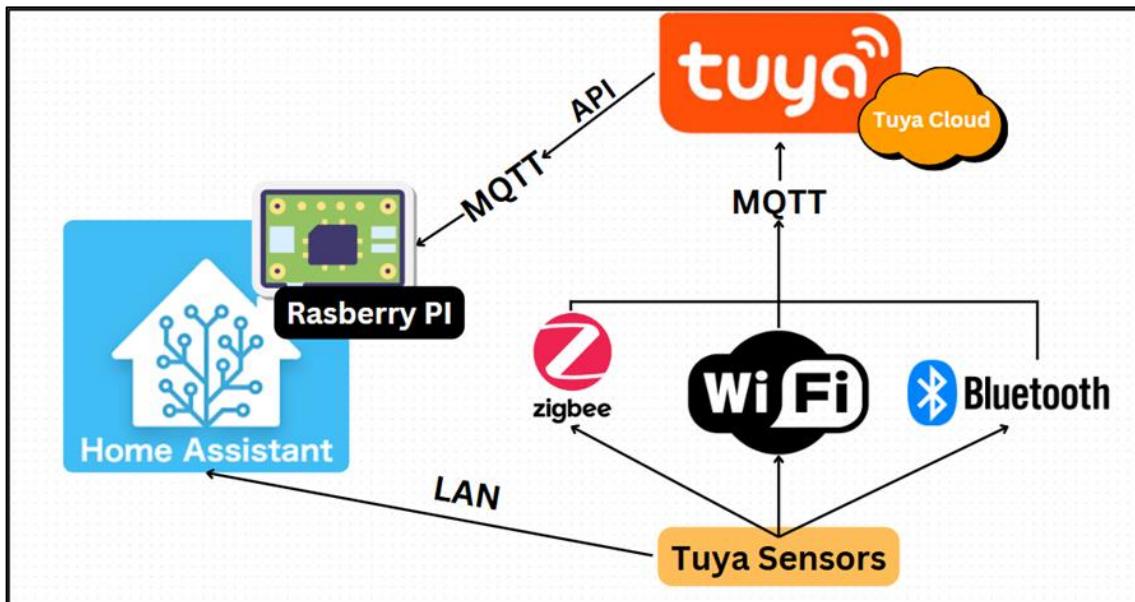
ESP Home ดังแสดงในรูปที่ 2.12 ESP home structure มีข้อดีหลายประการ เช่น มีการสร้างโค้ดควบคุมอุปกรณ์ผ่านการกำหนดค่าแบบ YAML ที่เป็นเรื่องง่ายและเข้าใจได้, มีการติดต่อสื่อสารกับอุปกรณ์ได้อย่างง่ายดายผ่านการใช้โปรโตคอล MQTT หรือ API, และมีการสนับสนุนอุปกรณ์ต่าง ๆ ที่หลากหลาย ซึ่งทำให้ ESP Home เป็นเครื่องมือที่ยอดเยี่ยมสำหรับผู้ที่ต้องการสร้างโปรเจกต์ IoT ในบ้านหรืออื่น ๆ อีกมากมาย โดยไม่ต้องมีความเชี่ยวชาญในการเขียนโปรแกรมอย่างลึกซึ้ง



รูปที่ 2.12 ESP home structure

2.1.6.3 Tuya

Tuya คือบริษัทที่มีตัวกลางในการพัฒนาแพลตฟอร์มสำหรับอินเทอร์เน็ตของสิ่งของ (IoT) ที่ช่วยให้ผู้ใช้สามารถเชื่อมต่อและควบคุมอุปกรณ์ต่าง ๆ สามารถทำงานกับ Home assistant ดังแสดงในรูปที่ 2.13 Tuya architecture ในบ้านหรือธุรกิจผ่านเครือข่ายอินเทอร์เน็ตได้อย่างง่ายดาย โดยใช้แพลตฟอร์มและเทคโนโลยีที่พัฒนาขึ้นโดยเฉพาะ สินค้าและบริการของ Tuya มักจะใช้ในการสร้างอุปกรณ์อัจฉริยะ เช่น หลอดไฟ LED, ประตูอัตโนมัติ, สมาร์ทโทรศัพท์, กล้องวงจรปิดเครื่องปรับอากาศ และอุปกรณ์อื่น ๆ ซึ่งสามารถควบคุมผ่านแอปพลิเคชันบนสมาร์ทโฟนหรืออุปกรณ์อื่น ๆ ที่เชื่อมต่อกับอินเทอร์เน็ตได้โดย Tuya ยังมีพันธมิตรหลายรายที่ผลิตและจำหน่ายอุปกรณ์ที่ใช้แพลตฟอร์ม



รูปที่ 2.13 Tuya architecture

2.1.6.4 Application tools

ReactJS คือไลบรารี JavaScript ที่พัฒนาโดย Meta (เดิมคือ Facebook) เพื่อสร้างส่วนติดต่อผู้ใช้ (UI) ที่มีความยืดหยุ่นและประสิทธิภาพสูง โดย React จะมุ่งเน้นไปที่การสร้างส่วนประกอบที่สามารถนำกลับมาใช้ใหม่ได้ ซึ่งสามารถนำไปใช้ในการพัฒนาเว็บแอปพลิเคชันที่มีโครงสร้างชั้นช้อน React ได้รับความนิยมอย่างมากเนื่องจากแนวทางในการพัฒนาแบบ “Component-Based Architecture” ที่ทำให้การพัฒนาเว็บมีความง่ายดาย ยืดหยุ่น และสามารถจัดการได้ง่ายในโครงการขนาดใหญ่

Component-Based Architecture React ทำงานโดยการแบ่งแอปพลิเคชันออกเป็นส่วนเล็ก ๆ ที่เรียกว่า "Component" ซึ่ง Component เป็นส่วนประกอบย่อยของ UI ที่สามารถนำกลับมาใช้ใหม่ได้ตัวอย่างเช่น มีปุ่มหนึ่งปุ่มที่ถูกสร้างขึ้นในรูปแบบของ Component สามารถใช้ปุ่มนั้นได้ในหลาย ๆ ที่ในแอปพลิเคชันเพียงแค่เรียกใช้งาน Component เดิมนั้น โดยที่ไม่ต้องสร้างใหม่ทุกครั้ง การออกแบบลักษณะนี้ช่วยให้การจัดการโค้ดง่ายขึ้นลดการทำงานที่ซ้ำซ้อน และทำให้การพัฒนาโค้ดมีประสิทธิภาพมากยิ่งขึ้น

React ใช้เทคโนโลยีที่เรียกว่า Virtual DOM (Document Object Model) เพื่อทำให้การอัปเดต UI มีประสิทธิภาพมากยิ่งขึ้น แทนที่จะทำการอัปเดต DOM จริงทั้งหมด React จะสร้าง Virtual DOM ขึ้นมา ซึ่งเป็นตัวแทนของ DOM จริง ๆ และทำการเปรียบเทียบ (diffing) กับ Virtual DOM เวอร์ชันก่อนหน้า จากนั้น React จะทำการอัปเดตเฉพาะส่วนที่มีการเปลี่ยนแปลงเท่านั้น ทำให้กระบวนการนี้รวดเร็วกว่า การอัปเดต DOM จริงในเว็บเบราว์เซอร์นั้นมีค่าใช้จ่ายสูงในแง่ของประสิทธิภาพ

ใน React มีสิ่งที่เรียกว่า JSX ซึ่งเป็นส่วนขยายของ JavaScript ที่ช่วยให้นักพัฒนาสามารถเขียนโค้ดในรูปแบบที่ดูเหมือน HTML ภายใน JavaScript ได้โดยตรง JSX ทำให้การพัฒนา UI ง่ายขึ้น เพราะสามารถเขียนโครงสร้างแบบ HTML ได้โดยตรงภายในโค้ดของ Component โดยไม่ต้องแยกส่วนระหว่าง JavaScript และ HTML อย่างในแนวทางดังเดิม

TailwindCSS เป็นเฟรมเวิร์ก CSS ที่ออกแบบมาเพื่อช่วยให้นักพัฒนาสามารถจัดการกับการออกแบบส่วนติดต่อผู้ใช้ (UI) ได้ง่ายและรวดเร็วมากขึ้น โดยมุ่งเน้นไปที่การใช้คลาสแบบ utility-first หรือคลาสที่ถูกสร้างมาเพื่อการจัดการสไตล์เฉพาะแบบทันที Utility class เหล่านี้ทำให้มีจำเป็นต้องสร้าง CSS เองในไฟล์แยก แต่สามารถจัดการสไตล์ทั้งหมดได้จากไฟล์ HTML โดยตรง ช่วยเพิ่มความสะดวกในการพัฒนาและลดความซับซ้อนในการจัดการสไตล์ในโครงการที่มีขนาดใหญ่

มีการใช้แนวคิด utility-first CSS framework ที่หมายถึงการจัดการการออกแบบด้วยคลาสที่มีพังก์ชันเฉพาะ ทำให้เราสามารถเพิ่มคลาสสำหรับสไต์ล์ต่าง ๆ เช่น ระยะขอบ สี ขนาด หรือแบบอักษร ได้โดยตรงภายในองค์ประกอบ HTML การออกแบบแบบ utility-first จะช่วยให้นักพัฒนามีความยืดหยุ่นสูงในการปรับแต่ง UI โดยไม่ต้องสร้าง CSS ใหม่และลดการเขียนโค้ดที่ซ้ำซ้อน

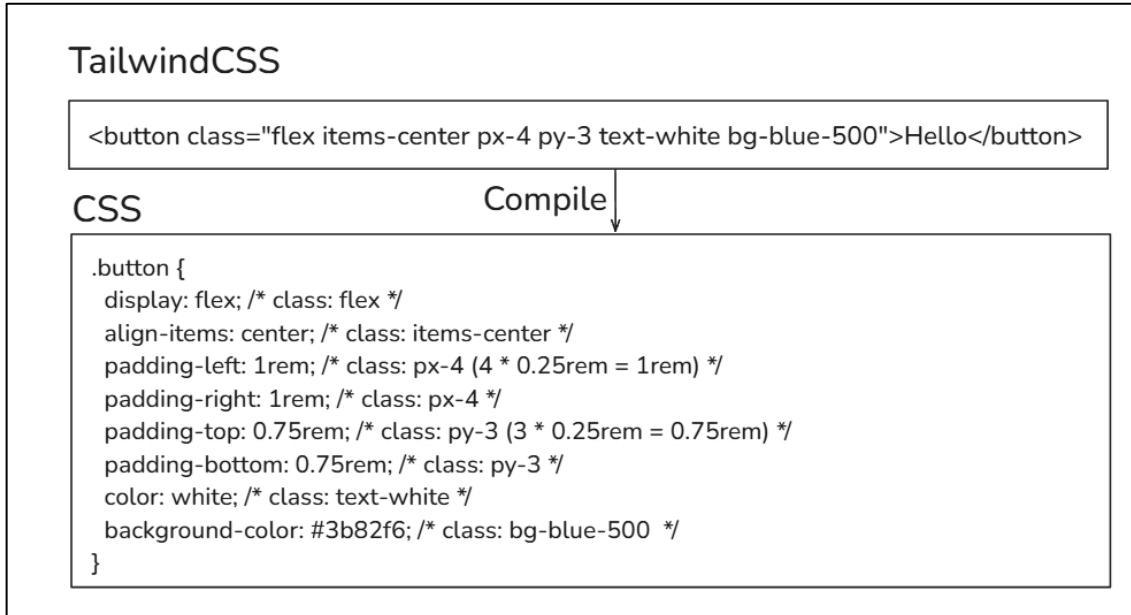
การออกแบบที่ยืดหยุ่น TailwindCSS ช่วยให้นักพัฒนาสามารถออกแบบและปรับแต่งส่วนต่าง ๆ ของ UI ได้อย่างอิสระและง่ายดาย โดยไม่จำเป็นต้องยึดตามเทมเพลตสำเร็จรูป นอกจากนี้ ยังมีระบบที่รองรับการปรับแต่งขนาด สี และระยะห่างต่าง ๆ ได้ตามความต้องการ โดยใช้ไฟล์ tailwind.config.js ในการตั้งค่าปรับแต่งเพิ่มเติม

การรองรับ Responsive Design TailwindCSS รองรับการออกแบบที่ปรับเปลี่ยนได้ตามขนาดหน้าจอ (responsive design) ด้วยการใช้คลาสต่าง ๆ ที่ตั้งค่ามาให้เรียกใช้งานง่าย เช่น sm, md, lg, xl และ 2xl เพื่อกำหนดสไต์ล์ตามขนาดหน้าจอที่แตกต่างกันได้อย่างง่ายดาย

Just-in-Time (JIT) Mode ในเวอร์ชันใหม่ ๆ ของ TailwindCSS ได้เพิ่มโหมด JIT (Just-in-Time) ที่ช่วยเพิ่มประสิทธิภาพในการทำงาน โดยสร้างเฉพาะ CSS ที่ถูกใช้งานในโปรเจกต์จริง ๆ เท่านั้น ทำให้ขนาดของไฟล์ CSS มีขนาดเล็กลง และทำให้โหลดเร็วขึ้น JIT Mode ยังรองรับการใช้งาน utility class แบบไดนามิก (dynamic utility class)

การใช้งานร่วมกับเฟรมเวิร์กและไลบรารีอื่น ๆ TailwindCSS สามารถทำงานร่วมกับเฟรมเวิร์ก หรือไลบรารี JavaScript และ CSS อื่น ๆ ได้อย่างมีประสิทธิภาพ ไม่ว่าจะเป็น React, Vue.js, Angular หรือ Svelte TailwindCSS ช่วยให้นักพัฒนาสามารถนำมาระบบยุกต์ใช้กับโครงการที่มีอยู่แล้วได้อย่างราบรื่น โดยไม่จำเป็นต้องเปลี่ยนโครงสร้างทั้งหมดของโครงการ

ไม่มี CSS ที่ไม่ถูกใช้งาน (Unused CSS) TailwindCSS จะสร้างเฉพาะ CSS ที่ถูกใช้งานในโปรเจกต์จริง ๆ และลบคลาสที่ไม่ได้ถูกเรียกใช้ออกไป ซึ่งช่วยให้ไฟล์ CSS ที่ได้มีขนาดเล็กลงมาก และทำให้การโหลดเว็บไซต์เร็วขึ้น ซึ่งเป็นจุดเด่นที่แตกต่างจากเฟรมเวิร์กอื่น ๆ ที่มีการนำเสนocomponent สำเร็จรูปแต่ไม่ได้นำไปใช้จริงทั้งหมด



รูปที่ 2.14 การคอมไพล์ CSS ด้วย Tailwind

Shadcn UI ในยุคที่เทคโนโลยีก้าวหน้าอย่างรวดเร็ว การพัฒนาเว็บแอปพลิเคชันที่มีประสิทธิภาพ และตอบสนองความต้องการของผู้ใช้เป็นสิ่งสำคัญ ซึ่งการออกแบบส่วนติดต่อผู้ใช้ (UI) เป็นหนึ่งในปัจจัยที่มีผลต่อประสบการณ์ของผู้ใช้ที่ดี การสร้าง UI ที่สวยงามและสามารถเข้าถึงได้จะเป็นความท้าทายสำหรับนักพัฒนา ในบทความนี้เราจะพูดถึง Shadcn UI ซึ่งเป็นไลบรารี UI สำหรับ React ที่มุ่งหวังในการสร้างอินเทอร์เฟซผู้ใช้ที่สวยงาม ตอบสนองได้ดี และมีความยืดหยุ่น

Shadcn UI ถูกสร้างขึ้นเพื่อช่วยให้การพัฒนา UI เป็นเรื่องง่ายและรวดเร็ว โดยมีแนวคิดที่มุ่งเน้นการให้ความสำคัญกับการออกแบบที่ทันสมัยและการปรับแต่งที่ง่าย ไลบรารีนี้ทำงานร่วมกับ Tailwind CSS ซึ่งเป็นเครื่องมือที่ได้รับความนิยมในกลุ่มนักพัฒนา โดยมอบความยืดหยุ่นในการจัดการสไต์ล์และการออกแบบให้กับนักพัฒนา

ส่วนประกอบที่ออกแบบมาอย่างดี Shadcn UI มีคอลเลกชันของส่วนประกอบที่ออกแบบมาอย่างดี เช่น ปุ่ม (Button), กล่องข้อความ (Input), เมนู (Menu) และอื่นๆ ซึ่งสามารถนำไปใช้ได้ทันทีโดยไม่ต้องสร้างสไต์ล์ใหม่จากศูนย์

การสนับสนุน Tailwind CSS Shadcn UI มีการออกแบบที่สามารถทำงานร่วมกับ Tailwind CSS ได้อย่างราบรื่น ทำให้นักพัฒนาสามารถใช้ utility classes ของ Tailwind ในการปรับแต่งส่วนประกอบได้ตามต้องการ

ความยืดหยุ่นในการปรับแต่ง นักพัฒนาสามารถปรับแต่งลักษณะและพฤติกรรมของส่วนประกอบได้ง่ายๆ ผ่าน props ที่ให้ว่า ทำให้การสร้าง UI ที่มีลักษณะเฉพาะตัวสามารถทำได้ง่ายขึ้น

การเข้าถึง (Accessibility) Shadcn UI มุ่งเน้นให้ความสำคัญกับการเข้าถึง เพื่อให้แน่ใจว่าผู้ใช้ทุกคนสามารถใช้งานได้ง่าย รวมถึงผู้ใช้ที่มีความต้องการพิเศษ

การตอบสนอง (Responsive Design) ไลบรารีนี้ออกแบบมาให้สามารถทำงานได้บนอุปกรณ์ที่หลากหลาย เช่น スマートフォン, แท็บเล็ต, และเดสก์ท็อป

เอกสารและการสนับสนุนที่ดี Shadcn UI มีเอกสารที่ชัดเจนและละเอียด ซึ่งช่วยให้นักพัฒนาสามารถเริ่มต้นใช้งานได้ง่ายและเรียนรู้วิธีการใช้งานฟีเจอร์ต่างๆ ได้อย่างมีประสิทธิภาพ

Rspack ในปัจจุบัน การพัฒนาเว็บแอปพลิเคชันมีความซับซ้อนมากขึ้นเรื่อยๆ เนื่องจากการเติบโตของเทคโนโลยีและการเพิ่มความคาดหวังของผู้ใช้ที่ต้องการความเร็วในการโหลดหน้าเว็บและประสบการณ์การใช้งานที่ราบรื่น การสร้างและจัดการโมดูลจึงกลายเป็นขั้นตอนที่มีความสำคัญในกระบวนการพัฒนาเว็บ เครื่องมือสำหรับการจัดการและบันเดลไฟล์อย่างเช่น Webpack และ Vite เป็นที่นิยมในกลุ่มนักพัฒนา อย่างไรก็ตาม เทคโนโลยีใหม่นี้ยังสามารถพัฒนาให้เร็วขึ้นและประสิทธิภาพดีขึ้น ได้ Rspack เป็นตัวอย่างหนึ่งของเครื่องมือที่ถูกพัฒนาขึ้นมาเพื่อตอบสนองความต้องการในการเพิ่มประสิทธิภาพการสร้างและการพัฒนาเว็บแอปพลิเคชัน

Rspack เป็นเครื่องมือจัดการและบันเดลเว็บแอปพลิเคชันที่ถูกออกแบบมาเพื่อให้ความเร็วและประสิทธิภาพสูงสุด มันเป็นทางเลือกใหม่ที่มุ่งเน้นความเร็วในการสร้างไฟล์บันเดล (bundle) และกระบวนการพัฒนาแบบเรียลไทม์ รวมถึงการโหลดซ้ำ (hot reloading) ที่รวดเร็วขึ้น โดยมีเป้าหมายให้การทำงานง่ายและรวดเร็วกว่าเครื่องมืออื่นๆ ในตลาด เช่น Webpack

ความเร็วในการสร้างบันเดล Rspack ถูกออกแบบมาให้ใช้เวลาน้อยที่สุดในการบันเดลโมดูลต่างๆ ในแอปพลิเคชัน โดยใช้เทคนิคการจัดการไฟล์ที่ทันสมัย ช่วยลดเวลาในการประมวลผลของไฟล์ที่มีขนาดใหญ่ หรือมีความซับซ้อนสูง ทำให้เหมาะสมสำหรับโครงการที่ต้องการประสิทธิภาพในขั้นตอนการสร้าง (build) และการพัฒนา

การรองรับการพัฒนาแบบเรียลไทม์ (Real-time Development) หนึ่งในจุดเด่นของ Rspack คือการรองรับการพัฒนาแบบทันที ทำให้นักพัฒนาเห็นการเปลี่ยนแปลงของโค้ดที่ทำแบบเรียลไทม์โดยไม่ต้องรอการบันเดลใหม่ทั้งหมด ความสามารถนี้ช่วยลดเวลาในการพัฒนาและเพิ่มประสบการณ์การใช้งานที่ดีขึ้น

การจัดการโมดูลที่มีประสิทธิภาพ Rspack มีความสามารถในการจัดการโมดูลอย่างมีประสิทธิภาพ โดยสามารถประมวลผลไฟล์ที่มีความซับซ้อนและไฟล์หลายประเภทได้อย่างรวดเร็ว รวมถึงการประมวลผลไฟล์ขนาดใหญ่ได้โดยไม่กระทบกับประสิทธิภาพของระบบ

การรองรับ Plugin เช่นเดียวกับ Webpack Rspack รองรับการใช้งานปลั๊กอินที่หลากหลาย เช่น ปลั๊กอินสำหรับการจัดการไฟล์ CSS, การจัดการภาพ และการบีบอัดข้อมูล ซึ่งสามารถช่วยให้กระบวนการสร้างไฟล์มีความยืดหยุ่นมากขึ้น โดยนักพัฒนาสามารถเพิ่มฟังก์ชันการทำงานเพิ่มเติมได้ตามความต้องการ

การตั้งค่าที่ง่ายและยืดหยุ่น Rspack มีการตั้งค่าที่เรียบง่ายและไม่ซับซ้อน สามารถใช้งานได้ทันทีโดยไม่ต้องมีการตั้งค่าที่ซับซ้อนเหมือนเครื่องมืออื่นๆ ซึ่งเป็นประโยชน์สำหรับนักพัฒนาที่ต้องการเริ่มต้นใช้งานอย่างรวดเร็ว

2.1.6.5 Nodejs

Node.js เป็นแพลตฟอร์มที่ใช้ในการพัฒนาแอปพลิเคชันฝั่งเซิร์ฟเวอร์ที่ทำงานด้วย JavaScript ซึ่งเดิมที่ JavaScript เป็นภาษาที่ใช้งานบนเว็บเบราว์เซอร์สำหรับการพัฒนาฝั่งคลาวน์เท่านั้น แต่ Node.js ทำให้เราสามารถใช้ JavaScript ใน การพัฒนาแอปพลิเคชันที่ทำงานบนเซิร์ฟเวอร์ได้ด้วย คุณสมบัติของ Node.js คือเป็น รันไทม์แบบ Cross-platform Node.js สามารถทำงานได้บนระบบปฏิบัติการหลากหลาย เช่น Windows, Linux, macOS Non-blocking I/O Node.js ใช้การทำงานแบบ asynchronous ซึ่งช่วยให้สามารถจัดการการเรียกใช้ข้อมูลหลายๆ คำขอ (requests) ได้ในเวลาเดียวกันโดยไม่ต้องรอคำขอ ก่อนหน้าเสร็จสิ้น (non-blocking) ใช้ได้กับ JavaScript Node.js ใช้ภาษา JavaScript ซึ่งทำให้นักพัฒนาที่คุ้นเคยกับ JavaScript สามารถพัฒนาแอปพลิเคชันฝั่งเซิร์ฟเวอร์ได้ทันที Event-driven Node.js ใช้การทำงานแบบ event-driven ซึ่งหมายความว่าการทำงานของระบบจะถูกกระตุ้นโดยเหตุการณ์ต่างๆ เช่น การรับคำขอจากผู้ใช้หรือการโหลดข้อมูลจากฐานข้อมูล เพ็กเกจจัดการง่าย (NPM) Node.js มาพร้อมกับตัวจัดการแพ็กเกจที่เรียกว่า npm (Node Package Manager) ซึ่งช่วยให้นักพัฒนาสามารถติดตั้งและใช้แพ็กเกจต่างๆ ได้ง่าย

2.1.6.6 Visual studio code

Visual Studio Code (VS Code) เป็นโปรแกรมแก้ไขโค้ด (code editor) แบบโอเพ่นซอร์สที่พัฒนาโดย Microsoft ซึ่งออกแบบมาเพื่อรองรับการพัฒนาโปรแกรมและเขียนโค้ดในหลากหลายภาษา เช่น JavaScript, Python, C++, Java, และอีกมากมาย มันได้รับความนิยมอย่างสูงในหมู่นักพัฒนานึ่งจากมีคุณสมบัติครบครัน, ใช้งานง่าย, และสามารถขยายความสามารถได้ผ่านทาง ส่วนขยาย (extensions) คุณสมบัติเด่นของ Visual Studio Code คือ อินเทอร์เฟซที่ใช้งานง่าย VS Code มีหน้าตาที่เรียบง่ายแต่ทรงพลัง มีแดบด้านข้างสำหรับการจัดการไฟล์และโฟลเดอร์ รวมถึงหน้าต่างหลักสำหรับเขียนโค้ด พร้อมด้วยการแสดงสีและการไฮไลท์โค้ดตามภาษาที่ใช้ รองรับหลายภาษา VS Code รองรับการเขียนโค้ดในหลายภาษา (Programming Languages) โดยอัตโนมัติ และมีส่วนขยายเพิ่มเติมสำหรับการรองรับภาษาใหม่ ๆ ที่อาจไม่ได้มีในตัวโปรแกรมมาตั้งแต่ต้น VS Code มีตลาดสำหรับส่วนขยายที่ทำให้นักพัฒนาสามารถเพิ่มฟีเจอร์พิเศษได้ เช่น เครื่องมือดีบัก, เทมเพลตสำหรับภาษาโปรแกรมเฉพาะ, อีม, และเครื่องมือสำหรับการเชื่อมต่อ กับระบบควบคุมเวอร์ชันอย่าง Git IntelliSense (คำแนะนำนำโค้ด)ฟีเจอร์นี้ช่วยในการเติมคำโค้ดอัตโนมัติ พร้อมแนะนำฟังก์ชันและไวยากรณ์ที่ถูกต้อง ทำให้ง่ายต่อการเขียนโค้ด และลดความผิดพลาด การดีบัก (Debugging) VS Code มีเครื่องมือดีบักที่ใช้งานง่าย ซึ่งช่วยให้สามารถตรวจสอบข้อผิดพลาดในโค้ดได้โดยตรงภายในโปรแกรมโดยไม่ต้องใช้เครื่องมือภายนอก Git Integration มีการเชื่อมต่อ กับระบบควบคุมเวอร์ชันอย่าง Git อย่างสมบูรณ์ สามารถทำงานกับ Git repository ได้โดยตรงจากโปรแกรม เช่น การ commit, push, pull, และดูประวัติการเปลี่ยนแปลงของโค้ด การทำงาน

ข้ามแพลตฟอร์ม VS Code สามารถทำงานได้บนระบบปฏิบัติการหลักทั้งสาม ได้แก่ Windows, macOS, และ Linux ซึ่งทำให้มันเป็นเครื่องมือที่ยืดหยุ่นมากในการใช้งานบนแพลตฟอร์มที่หลากหลาย การปรับแต่ง สามารถปรับแต่งการทำงานของ VS Code ได้อย่างอิสระตามความต้องการ ไม่ว่าจะเป็นการปรับแต่งหน้าต่างเครื่องมือหรือสร้างคีย์ลัดส่วนตัว รวมถึงการติดตั้งรีมเพื่อเปลี่ยนหน้าตาโปรแกรมให้ตรงกับความชอบ

2.1.6.7 Grafana K6

Grafana k6 เป็นเครื่องมือทดสอบโหลด (load testing) แบบโอลูเอนชอร์สที่พัฒนาโดย k6 (ปัจจุบันเป็นส่วนหนึ่งของ Grafana Labs) ถูกออกแบบมาเพื่อช่วยนักพัฒนาและผู้ดูแลระบบในการทดสอบสมรรถนะของแอปพลิเคชันและ API ในสภาพแวดล้อมการใช้งานจริง เครื่องมือนี้เน้นการทดสอบการทำงานของระบบภายใต้สภาวะการใช้งานหนัก (heavy load) เช่น การจำลองผู้ใช้งานจำนวนมากที่เข้าถึงแอปพลิเคชันพร้อมกัน เพื่อวัดว่าระบบสามารถรองรับการทำงานได้อย่างไร้ปัญหาหรือไม่

Grafana k6 ทำให้การสร้างและจัดการสคริปต์ทดสอบทำได้ง่ายโดยใช้ JavaScript เพื่อกำหนดการทดสอบการเรียก API และพฤติกรรมของผู้ใช้ การเขียนสคริปต์ที่ยืดหยุ่นนี้ช่วยให้สามารถจำลองสถานการณ์ที่ซับซ้อน เช่น การล็อกอิน การส่งข้อมูล การอัปเดตไฟล์ หรือการทำงานแบบต่อเนื่องในระบบ โดยสามารถควบคุมปริมาณผู้ใช้ที่เข้าสู่ระบบ (concurrent users) และการตั้งค่าวงจรการทดสอบ (test duration) ได้

Virtual Users (VUs) เป็นองค์ประกอบสำคัญใน Grafana k6 ที่ใช้ในการจำลองผู้ใช้งานหลายคนพร้อมกันในการทดสอบโหลด VUs เป็นตัวแทนของผู้ใช้งานที่มีปฏิสัมพันธ์กับระบบหรือ API ที่เรากำลังทดสอบ ในการทดสอบนั้นแต่ละ VU จะทำงานตามสคริปต์ที่เขียนขึ้นเพื่อจำลองการใช้งานจริง เช่น การเรียก API, การคลิกปุ่ม หรือการส่งแบบฟอร์มในเว็บแอปพลิเคชัน การทดสอบด้วย VUs ช่วยให้เราเข้าใจว่าแอปพลิเคชันจะทำงานอย่างไรเมื่อมีผู้ใช้งานจำนวนมากเข้ามาใช้งานพร้อมกัน

2.1.7 Data Communication

2.1.7.1 Restful API

RESTful API (Representational State Transfer Application Programming Interface) เป็นรูปแบบการออกแบบ API ที่ใช้แนวคิดของ REST ซึ่งเป็นสถาปัตยกรรมที่เน้นการใช้โปรโตคอล HTTP ในการสื่อสารระหว่างคลาวน์และเซิร์ฟเวอร์ แต่ละทรัพยากร เช่น ข้อมูล ผู้ใช้ หรือไฟล์ จะถูกเข้าถึงผ่าน URL หรือ URI และใช้ HTTP methods เช่น GET, POST, PUT, DELETE ในการทำงานกับทรัพยากรเหล่านี้ โดย RESTful API มักใช้การส่งข้อมูลแบบ JSON หรือ XML ในการแลกเปลี่ยนข้อมูล ทำให้มีการทำงานที่ง่ายและยืดหยุ่น เหมาะสมสำหรับการพัฒนาแอปพลิเคชันเว็บหรือบริการที่ต้องการการทำงานระหว่างเครื่องมือหรือระบบต่าง ๆ

ข้อดีของ RESTful API คือความง่ายในการใช้งานและความสามารถในการปรับตัวให้เข้ากับระบบที่มีอยู่ ด้วยการใช้ HTTP โปรโตคอลที่แพร่หลาย การสื่อสารระหว่างระบบจึงมีประสิทธิภาพและสามารถขยายได้อย่างยืดหยุ่น นอกจากนี้ RESTful API ยังเป็นสถาปัตยกรรมที่ไม่ต้องเก็บสถานะ (stateless) ซึ่ง

หมายความว่าการร้องขอแต่ละครั้งจะถูกประมวลผลอย่างเป็นอิสระจากกัน ทำให้การจัดการข้อมูลและการขยายระบบทำได้ง่ายขึ้น

2.1.7.2 Websocket API

WebSocket API เป็นโปรโตคอลที่ช่วยให้การสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์เป็นแบบสองทาง (full-duplex) ผ่านการเชื่อมต่อ TCP ที่คงอยู่ ซึ่งต่างจาก RESTful API ที่ทำงานแบบ stateless โดย WebSocket API จะทำให้สามารถส่งข้อมูลได้ทันทีทั้งสองฝ่ายหลังจากเชื่อมต่อสำเร็จ โดยไม่ต้องมีการสร้างการเชื่อมต่อใหม่สำหรับแต่ละการร้องขอ การใช้งานนี้เหมาะสมสำหรับแอปพลิเคชันที่ต้องการอัปเดตข้อมูลอย่างต่อเนื่องและทันที เช่น การแชทออนไลน์ เกมแบบเรียลไทม์ หรือการเทรดหุ้นแบบเรียลไทม์

การเชื่อมต่อ WebSocket เริ่มต้นด้วย HTTP handshake และเมื่อการเชื่อมต่อสำเร็จ จะเปลี่ยนเป็นโปรโตคอล WebSocket ที่ช่วยให้ส่งข้อมูลไปมาได้โดยไม่ต้องปิดการเชื่อมต่อ เมื่อไคลเอนต์หรือเซิร์ฟเวอร์ต้องการส่งข้อมูล พากษาสามารถส่งได้ทันทีโดยไม่ต้องร้องขอใหม่ การทำงานนี้ทำให้ WebSocket API มีประสิทธิภาพสูงและเหมาะสมกับงานที่ต้องการการโต้ตอบแบบทันที

2.1.7.3 MQTT

MQTT (Message Queuing Telemetry Transport) เป็นโปรโตคอลการรับส่งข้อมูลที่ถูกออกแบบมาเพื่อการสื่อสารระหว่างอุปกรณ์ที่มีทรัพยากรจำกัด เช่น อุปกรณ์ IoT (Internet of Things) ซึ่งโปรโตคอลนี้ได้รับการออกแบบมาโดย IBM ในปี 1999 และต่อมาได้รับการพัฒนาเป็นมาตรฐานเปิดของ OASIS ในปี 2013 เพื่อเป็นทางเลือกที่มีประสิทธิภาพสำหรับการส่งข้อมูลในสภาพแวดล้อมที่มีข้อจำกัดในการใช้พลังงาน แบบวิดิธ และทรัพยากรการประมวลผล MQTT ทำงานบนโมเดล Publish/Subscribe ซึ่งเป็นสถาปัตยกรรมการสื่อสารที่ต่างจากโมเดล Request/Response (เช่น HTTP) อย่างชัดเจน การทำงานแบบ Publish/Subscribe ทำให้สามารถแยกผู้ส่งข้อมูล (Publisher) ออกจากผู้รับข้อมูล (Subscriber) ได้โดยมีตัวกลางเรียกว่า Broker ทำหน้าที่เป็นผู้จัดการการสื่อสาร Publisher อยู่ที่ตัวกลางเรียกว่า Broker ทำหน้าที่ในการส่งข้อมูลหรือข้อความออกไป ตัวอย่างเช่น เช่นเซอร์อุณหภูมิที่ส่งค่าที่วัดได้ไปยัง Broker Publisher ไม่ต้องรู้ว่าใครจะรับข้อมูลนี้ เพียงแค่ส่งข้อมูลไปยัง Broker เท่านั้น ข้อมูลที่ส่งจะถูกจัดในรูปแบบของ Topic ซึ่งเป็นเส้นทางที่กำหนดว่าข้อมูลนั้นจะเกี่ยวข้องกับเรื่องอะไร เช่น "home/temperature" อาจใช้เป็น Topic สำหรับส่งข้อมูลอุณหภูมิในบ้าน อุปกรณ์หรือระบบที่สมัครสมาชิก (subscribe) เพื่อรับข้อมูลจาก Topic ที่สนใจ ตัวอย่างเช่น ระบบควบคุมอุณหภูมิอัตโนมัติอาจจะสมัครสมาชิกกับ Topic "home/temperature" เพื่อรับข้อมูลอุณหภูมิแล้วทำงานตามค่าที่ได้รับ Subscriber สามารถเลือกสมัครสมาชิกกับหลายๆ Topic หรือเฉพาะบาง Topic ที่ต้องการได้ Broker เป็นเซิร์ฟเวอร์ที่ทำหน้าที่เป็นตัวกลางในการรับส่งข้อมูลระหว่าง Publisher และ Subscriber โดยจะทำหน้าที่รับข้อมูลจาก Publisher และส่งต่อให้ Subscriber ที่สมัครสมาชิกกับ Topic นั้นๆ Broker มีหน้าที่จัดการการเชื่อมต่อ การกระจายข้อมูล และตรวจสอบสิทธิ์การเข้าถึง โดยในบางกรณีอาจมีการใช้ระบบรักษาความปลอดภัยเช่น TLS เพื่อป้องกันการเข้าถึงข้อมูลที่ไม่ได้รับอนุญาต Topic คือเส้นทางหรือ

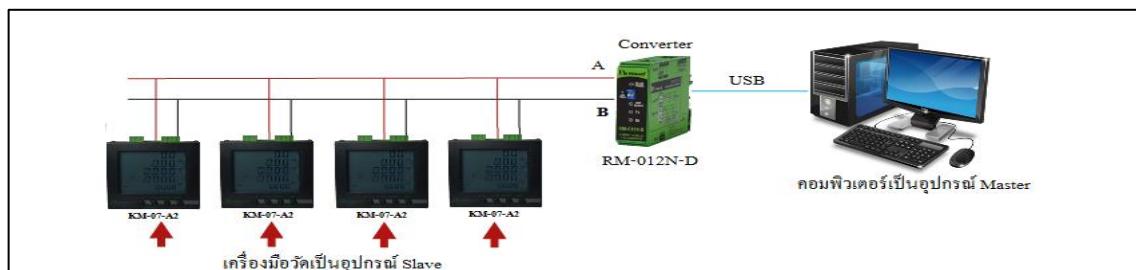
ที่อยู่ที่ใช้ในการจัดเก็บและส่งต่อข้อมูลระหว่าง Publisher และ Subscriber ตัวอย่างเช่น "sensor/livingroom/temperature" เป็น Topic ที่อาจใช้เพื่อส่งค่าของเซ็นเซอร์ในห้องนั่งเล่น การออกแบบ Topic ช่วยในการจัดกลุ่มและกรองข้อมูล ทำให้ Subscriber สามารถรับเฉพาะข้อมูลที่ต้องการได้โดยไม่จำเป็นต้องรับข้อมูลทั้งหมด โดยมี Message คือ ข้อมูลที่ส่งในโปรโตคอล MQTT จะถูกบรรจุอยู่ใน Message ซึ่งแต่ละ Message จะประกอบด้วย Topic, Payload (เนื้อหาของข้อความ), คุณสมบัติ QoS (Quality of Service) และอื่นๆ Message นี้จะถูกส่งจาก Publisher ไปยัง Broker และจากนั้น Broker จะกระจายไปยัง Subscriber ที่สนใจ

2.1.7.4 Recommended Standard no. 485

RS485 (ย่อมาจาก: Recommended Standard no. 485) คือมาตรฐานการสื่อสารข้อมูลดิจิตอลแบบอนุกรม (serial communication) ซึ่งถูกกำหนดขึ้นครั้งแรกในปี ค.ศ. 1998 โดยความร่วมมือของ TIA (Telecommunications Industry Association) และ EIA (Electronic Industries Association) มาตรฐาน RS485 ถูกใช้อย่างแพร่หลายในโรงงานอุตสาหกรรม เนื่องจากสามารถส่งสัญญาณได้ไกลและยังสามารถส่งพร้อมๆ กันได้หลายจุด

ปกติแล้ว EIA จะตั้งชื่อมาตรฐานของตัวเองโดยการใช้คำนำหน้าว่า "RS" (Recommended Standard) แต่เนื่องจากมาตรฐานนี้เป็นความร่วมมือระหว่าง 2 หน่วยงาน คือ TIA และ EIA ทั้งสองหน่วยงานจึงตกลงเปลี่ยนจากคำว่า "RS" เป็น "TIA/EIA" แทนอย่างเป็นทางการ เพื่อระบุถึงแหล่งที่มาของมาตรฐานอย่างชัดเจน โดยต่อมาทาง EIA ที่ได้ยกเลิกมาตรฐานนี้และมาตรฐาน RS485 นี้ก็ได้ถูกพัฒนาอย่างต่อเนื่องจนถึงปัจจุบันโดย TIA ทำให้มาตรฐาน RS485 ถูกเปลี่ยนชื่อเป็น "TIA-485" อย่างเป็นทางการ แต่สุดท้ายเพราความเคยชินทำให้วิศวกรทั่วโลกยังเรียกมาตรฐานการสื่อสารนี้ว่า RS485 เหมือนเดิม

มาตรฐาน RS485 สามารถเชื่อมต่อการรับส่งข้อมูลแบบเครือข่าย (Network) โดยมีอุปกรณ์ในเครือข่ายได้สูงสุดถึง 32 ตัว ซึ่งในเครือข่ายนี้ จะต้องมีอุปกรณ์อยู่ 1 ตัว ทำหน้าที่คือผู้จัดการสื่อสาร ในเครือข่าย ซึ่งเราจะเรียกอุปกรณ์ตัวนี้ว่า "Master" และอุปกรณ์ส่วนที่เหลือเราจะเรียกว่า "Slave" โดยที่ Slave แต่ละตัวจะมีหมายเลข Address ของตัวเอง และเมื่อตัว Master ต้องการสั่งการตัว Slave ตัว Master จะส่งชุดคำสั่งพร้อมระบุหมายเลข Address ไปยังอุปกรณ์ Slave ทุกตัว เมื่ออุปกรณ์ Slave ได้รับคำสั่งและคำสั่งนั้นมีหมายเลข Address ตรงกับตัวเอง อุปกรณ์ Slave ถึงจะทำการคำสั่งของ Master เป็นลำดับไป



รูปที่ 2.15 การทำงานของ RS485 แบบ Network

2.1.7.5 Cloudflare

Cloudflare มีบริการหลายประเภทที่ครอบคลุมทุกด้านของการจัดการเครือข่าย การรักษาความปลอดภัย และการเพิ่มประสิทธิภาพ ซึ่งสามารถแบ่งได้เป็นสามกลุ่มหลักคือ Content Delivery Network (CDN), บริการรักษาความปลอดภัย และ บริการที่เกี่ยวข้องกับการจัดการ DNS โดยบริการเหล่านี้ทำงานร่วมกันเพื่อช่วยให้เว็บไซต์และบริการออนไลน์ทำงานได้อย่างมีประสิทธิภาพมากขึ้น ในขณะเดียวกันก็ป้องกันการโจมตีและภัยคุกคามต่างๆ ที่เกิดขึ้นบนอินเทอร์เน็ต

Content Delivery Network (CDN) หนึ่งในบริการที่สำคัญของ Cloudflare คือ Content Delivery Network (CDN) ซึ่งเป็นระบบเซิร์ฟเวอร์ที่กระจายตัวอยู่ทั่วโลก ทำหน้าที่จัดเก็บและส่งมอบเนื้อหาเว็บ (Content) ให้กับผู้ใช้งานจากตำแหน่งที่ใกล้ที่สุดทางภูมิศาสตร์ ซึ่งช่วยลดเวลาในการโหลดเว็บไซต์หรือแอปพลิเคชันลงอย่างมาก โดยปกติแล้ว การให้บริการเนื้อหาผ่านเซิร์ฟเวอร์ของ Cloudflare ที่กระจายอยู่ในหลายภูมิภาคนั้น จะลดความล่าช้า (Latency) ใน การเข้าถึงข้อมูล ส่งผลให้ผู้ใช้งานสามารถโหลดหน้าเว็บได้เร็วขึ้นและเสถียรมากขึ้น

CDN ของ Cloudflare ทำงานโดยการเก็บข้อมูลแบบ Static เช่น ภาพและไฟล์ที่ไม่เปลี่ยนแปลงบ่อยๆ ไว้ในเซิร์ฟเวอร์ที่ใกล้ผู้ใช้งานที่สุด เช่น หากมีผู้ใช้ในประเทศไทย Cloudflare ก็จะใช้เซิร์ฟเวอร์ที่อยู่ในประเทศไทยในการให้บริการข้อมูลที่ถูกแคชไว้ (Cache) แทนที่จะต้องส่งข้อมูลจากเซิร์ฟเวอร์ในสหรัฐอเมริกาหรือยุโรป ซึ่งนอกจากจะช่วยให้โหลดเว็บได้เร็วขึ้นแล้ว ยังช่วยลดภาระงานของเซิร์ฟเวอร์ต้นทางอีกด้วย

ระบบการรักษาความปลอดภัย (Security Services) Cloudflare ให้บริการทางด้านการรักษาความปลอดภัยสำหรับเว็บไซต์และเครือข่ายอินเทอร์เน็ตในหลายด้าน โดยหนึ่งในฟังก์ชันหลักคือการป้องกันการโจมตีแบบ Distributed Denial-of-Service (DDoS) ซึ่งเป็นการโจมตีที่มุ่งเน้นไปที่การทำให้บริการล่มหรือไม่สามารถตอบสนองผู้ใช้งานได้ผ่านการส่งคำขอจำนวนมหาศาลในระยะเวลาอันสั้น

DDoS Protection Cloudflare มีโครงสร้างพื้นฐานที่สามารถตรวจจับและป้องกันการโจมตี DDoS ได้ในระดับที่มีประสิทธิภาพสูง การโจมตี DDoS นั้นสามารถเกิดขึ้นได้ทั้งจากเครือข่ายบอตเน็ต (Botnet) ที่ประกอบด้วยเครื่องคอมพิวเตอร์หรืออุปกรณ์เชื่อมต่อจำนวนมากที่ถูกโจมตีแล้วควบคุมมาใช้ในการโจมตี ผ่านการส่งคำขอไปยังเซิร์ฟเวอร์เป้าหมายพร้อมกันในครั้งเดียว หรือจากการโจมตีที่มาจากแหล่งที่มาหลายที่ในเวลาเดียวกัน (Distributed)

โครงสร้างพื้นฐานของ Cloudflare มีเซิร์ฟเวอร์ที่สามารถกรองคำขอที่ไม่ปกติหรือคำขอที่อาจมาจากการโจมตี DDoS ออกໄປได้โดยอัตโนมัติ ทำให้เว็บไซต์หรือบริการออนไลน์ที่ใช้ Cloudflare สามารถป้องกันการโจมตีได้โดยที่ไม่ส่งผลกระทบต่อผู้ใช้งานที่แท้จริง

Web Application Firewall (WAF) Cloudflare ยังมีระบบ Web Application Firewall (WAF) ที่ช่วยกรองและป้องกันการโจมตีที่มุ่งเป้าไปที่ช่องโหว่ในแอปพลิเคชันเว็บ เช่น การโจมตีแบบ SQL Injection, Cross-Site Scripting (XSS) หรือการโจมตีที่พยายามเข้าถึงข้อมูลที่ไม่ควรเปิดเผย ระบบ

WAF นี้จะทำหน้าที่ตรวจสอบคำขอที่ส่งเข้ามายังเว็บเซิร์ฟเวอร์และป้องกันคำขอที่มีลักษณะต้องสงสัยหรือเป็นอันตราย

SSL/TLS Encryption อีกหนึ่งความสามารถของ Cloudflare คือการจัดการ SSL/TLS เพื่อเข้ารหัสข้อมูลที่ส่งระหว่างผู้ใช้งานกับเซิร์ฟเวอร์ ซึ่งช่วยให้การสื่อสารออนไลนมีความปลอดภัยมากขึ้น Cloudflare มีบริการ SSL Certificate พรีที่ช่วยให้ผู้ใช้งานสามารถเข้ารหัสข้อมูลด้วยการใช้ HTTPS ได้ง่ายดายและรวดเร็ว โดยไม่ต้องติดตั้งหรือจัดการใบรับรอง SSL เอง นอกจากนี้ Cloudflare ยังรองรับการเข้ารหัสแบบ Full SSL และ Full (Strict) SSL ซึ่งจะเข้ารหัสการสื่อสารระหว่าง Cloudflare และเซิร์ฟเวอร์ต้นทางด้วยเช่นกัน

DNS Management อีกหนึ่งบริการสำคัญของ Cloudflare คือ DNS Management หรือการจัดการระบบชื่อโดเมน (Domain Name System) ซึ่งเป็นระบบที่ช่วยแปลงชื่อโดเมน เช่น www.example.com ไปเป็น IP Address ของเซิร์ฟเวอร์ที่ใช้งานอยู่ Cloudflare ให้บริการ DNS ที่เร็วและเสถียร ทำให้การแปลงชื่อโดเมนเป็น IP Address สามารถทำได้อย่างรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น นอกจากนี้ Cloudflare ยังมีคุณสมบัติด้านความปลอดภัยที่ช่วยป้องกันการโจมตีแบบ DNS Spoofing หรือ DNS Cache Poisoning ที่มุ่งทำลายระบบ DNS ของเว็บไซต์ DNS ของ Cloudflare นั้นทำงานในรูปแบบของ Anycast ซึ่งหมายความว่าเมื่อผู้ใช้ทำการร้องขอชื่อโดเมน ระบบจะส่งคำขอไปยังเซิร์ฟเวอร์ที่ใกล้ที่สุดในเครือข่าย Cloudflare โดยอัตโนมัติ ส่งผลให้การแปลง DNS เป็นไปได้อย่างรวดเร็วและลดเวลาในการรอโหลดเว็บไซต์

Cloudflare Workers นอกเหนือจากการด้านความปลอดภัยและประสิทธิภาพ Cloudflare ยังมีบริการ Cloudflare Workers ซึ่งเป็นแพลตฟอร์มสำหรับการรันโค้ด JavaScript หรือ WebAssembly ที่ขอบเครือข่าย (Edge Computing) ซึ่งช่วยให้ผู้พัฒนาแอปพลิเคชันสามารถรันฟังก์ชันหรือโปรแกรมขนาดเล็กใกล้กับผู้ใช้งานมากขึ้น ทำให้สามารถตอบสนองได้เร็วกว่าเดิม Cloudflare Workers เป็นโซลูชันที่สามารถใช้สร้างแอปพลิเคชันที่ทำงานได้เร็วขึ้นและลด Latency ใน การตอบสนอง นอกจากนี้ยังสามารถใช้เพื่อสร้างแอปพลิเคชันที่ต้องการขยายขนาด (Scalability) โดยไม่ต้องกังวลเรื่องการจัดการเซิร์ฟเวอร์

Zero Trust Security Cloudflare ยังมีโซลูชันด้านความปลอดภัยที่เรียกว่า Zero Trust ซึ่งเป็นแนวคิดในการรักษาความปลอดภัยที่ไม่ไว้วางใจการเชื่อมต่อใดๆ ทั้งจากภายในและภายนอกองค์กร โดยทุกการเชื่อมต่อจะต้องได้รับการตรวจสอบและอนุญาตก่อนที่จะสามารถเข้าถึงแอปพลิเคชันหรือข้อมูลได้ บริการ Zero Trust ของ Cloudflare ประกอบด้วยการจัดการการเข้าถึง (Access Control) ที่ซับซ้อน การตรวจสอบตัวตนผู้ใช้งาน และการกรองข้อมูลที่ส่งเข้ามาอย่างเข้มงวด ช่วยให้องค์กรสามารถรักษาความปลอดภัยของข้อมูลและแอปพลิเคชันได้อย่างมีประสิทธิภาพมากขึ้น

2.1.8 การดำเนินการเชิงสถิติ

2.1.8.1 ค่าเฉลี่ย

ค่าเฉลี่ย (Average) เป็นแนวคิดทางสถิติที่สำคัญ ใช้เพื่อแสดงค่ากลางหรือค่าที่เป็นตัวแทนของกลุ่มข้อมูล ค่าเฉลี่ยสามารถช่วยให้เราทราบถึงแนวโน้มและลักษณะของข้อมูลในชุดข้อมูลต่าง ๆ เช่น คะแนนสอบ ผลการขาย หรือค่าทางเศรษฐกิจ ค่าเฉลี่ยมีหลายประเภท เช่น ค่าเฉลี่ยเลขคณิต (Arithmetic Mean), ค่าเฉลี่ยเลขซึ่งกำลัง (Geometric Mean), และค่าเฉลี่ยเลขหมุน (Harmonic Mean) แต่ในที่นี้เราจะพูดถึงค่าเฉลี่ยเลขคณิตเป็นหลัก เนื่องจากเป็นที่นิยมและใช้งานบ่อยที่สุด ค่าเฉลี่ยมักถูกนำมาใช้ในหลายบริบท เช่น เพื่อตรวจสอบผลการเรียนของนักเรียน โดยการคำนวณค่าเฉลี่ยคะแนนสอบ จะช่วยให้ผู้สอนเห็นภาพรวมของความสำเร็จของนักเรียนในรายวิชานั้น ๆ ในกรณีวิเคราะห์ผลการขาย สามารถใช้ค่าเฉลี่ยในการประเมินแนวโน้มการขายสินค้าในช่วงเวลาต่าง ๆ การวิจัย ใช้ในการรวบรวมข้อมูลเพื่อนำมาวิเคราะห์ โดยการหาค่าเฉลี่ยจะช่วยให้เห็นภาพรวมและแนวโน้มของข้อมูลได้ง่ายขึ้นในบางกรณี การใช้ค่าเฉลี่ยอาจไม่เหมาะสม เช่น ถ้าชุดข้อมูลมีค่าผิดปกติหรือมีการกระจายที่ไม่สมมาตร ในกรณีนี้ อาจพิจารณาใช้ค่าเฉลี่ยเลขซึ่งกำลัง (Geometric Mean) หรือค่าเฉลี่ยเลขหมุน (Harmonic Mean) แทน เพื่อให้ได้ผลลัพธ์ที่สะท้อนค่ากลางได้ดียิ่งขึ้นค่าเฉลี่ยเป็นเครื่องมือที่มีประโยชน์ในการวิเคราะห์ข้อมูลที่ช่วยให้เราเข้าใจลักษณะและแนวโน้มของข้อมูลได้ดียิ่งขึ้น อย่างไรก็ตาม ควรพิจารณาสภาพของข้อมูลก่อนที่จะเลือกใช้ค่าเฉลี่ยเป็นตัวแทนค่ากลาง เนื่องจากอาจมีข้อจำกัดในบางสถานการณ์ การทำความเข้าใจเกี่ยวกับค่าเฉลี่ยและวิธีการใช้ที่เหมาะสมจะเป็นสิ่งสำคัญในการวิเคราะห์ข้อมูลอย่างมีประสิทธิภาพ ดังสมการที่ 2.2

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{สมการที่ 2.2}$$

เมื่อ \bar{x} คือ ค่าเฉลี่ย

x_i คือ ค่าต่าง ๆ ในชุดข้อมูล

n คือ จำนวนค่าทั้งหมดในชุดข้อมูล

2.1.8.2 ส่วนเบี่ยงเบนมาตรฐาน

ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation) เป็นตัวชี้วัดทางสถิติที่ใช้เพื่อวัดความกระจายหรือความแปรปรวนของชุดข้อมูล ซึ่งแสดงถึงว่าข้อมูลในชุดนั้น ๆ กระจายตัวอยู่ห่างจากค่าเฉลี่ย (Mean) มากน้อยเพียงใด การคำนวณส่วนเบี่ยงเบนมาตรฐานช่วยให้เราเข้าใจถึงลักษณะของข้อมูลได้ดียิ่ง และสามารถนำไปใช้ในหลาย ๆ ด้าน เช่น การวิเคราะห์ผลการศึกษา การประเมินความเสี่ยงในธุรกิจ หรือการวิจัยทางวิทยาศาสตร์ ส่วนเบี่ยงเบนมาตรฐานจะบอกถึงระดับการกระจายของข้อมูล เมื่อข้อมูลมีการ

กระจายตัวมาก หมายความว่าข้อมูลมีค่าห่างจากค่าเฉลี่ยมาก ส่วนเบี่ยงเบนมาตรฐานจะมีค่ามากขึ้น ในทางกลับกัน ถ้าข้อมูลมีการกระจายตัวน้อย ค่าของส่วนเบี่ยงเบนมาตรฐานจะน้อยลง ดังสมการที่ 2.3

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}} \quad \text{สมการที่ 2.3}$$

เมื่อ σ คือ ส่วนเบี่ยงเบนมาตรฐานของประชากร

x_i คือ ค่าต่าง ๆ ในชุดข้อมูล

μ คือ ค่าเฉลี่ยของประชากร

N คือ จำนวนค่าทั้งหมดในประชากร

2.2 บททวนวรรณกรรมที่เกี่ยวข้อง

กระทรวงพัฒนา (2015) แผนอนุรักษ์พัฒนา EEP 2015 กำหนดเป้าหมายในการลดการใช้พัฒนาลง 30% ภายในปี 2579 เมื่อเทียบกับปี 2553 ซึ่งโรงเรียนเป็นกลุ่มอาคารที่ใช้พลังงานเป็นจำนวนมาก ดังนั้น แนวทางและมาตรการต่างๆ ที่ระบุไว้ในแผน EEP 2015 จึงสามารถนำมาใช้เพื่อลดการใช้พลังงานในโรงเรียนได้แผน EEP 2015 กำหนดให้มีการบังคับใช้มาตรฐานการอนุรักษ์พลังงานสำหรับอาคารควบคุม ซึ่งรวมถึงโรงเรียนด้วย โดยมาตรฐานได้แก่ การติดตั้งระบบควบคุมและติดตามการใช้พลังงาน, การใช้อุปกรณ์ไฟฟ้าและเครื่องใช้ไฟฟ้าที่มีประสิทธิภาพสูง, การปรับปรุงระบบแสงสว่างและระบบปรับอากาศให้มีประสิทธิภาพยิ่งขึ้น, การปรับปรุงระบบระบายอากาศและระบบทำความเย็น, การปรับปรุงระบบน้ำและระบบบำบัดน้ำเสีย

จิรพันธ์ พิมพลและคณะ (2023) ระบบตรวจสอบและควบคุมโรงเรียนปลูกผักแบบปิด ออกแบบและสร้างระบบควบคุม สภาพแวดล้อมสำหรับปลูกผักในโรงเรียนไฮโดรโปนิกส์ (hydroponics) แบบระบบปิด ด้วยระบบอัตโนมัติให้มีความเหมาะสมต่อการ เจริญเติบโตของผัก ด้วยการประมวลผลจากไมโครคอนโทรลเลอร์ สามารถตรวจสอบค่าสถานะต่าง ๆ ภายใน โรงเรียนผ่านแอปพลิเคชัน Blink บนสมาร์ทโฟนและผ่านหน้าจอแสดงผลแอลซีดีบริเวณ โรงเรียน นอกจากนี้ระบบยังสามารถควบคุมการทำงานของพัดลมระบายอากาศ หลอดไฟ LED และปั๊มพ่นหมอกได้โดยอัตโนมัติ ตามค่าอุณหภูมิ ความชื้น ที่กำหนด ระบบควบคุมสภาพแวดล้อมภายในโรงเรียนประกอบ ไปด้วยไมโครคอนโทรลเลอร์ (NodeMCU ESP8266) ซึ่งเป็นตัวควบคุม ระบบ โดยใช้เซ็นเซอร์ AM2315 วัดค่าอุณหภูมิและความชื้น pH sensor วัดค่าความเป็นกรด-เบสในน้ำและใช้เซ็นเซอร์ Ultrasonic HC-SR04 วัดระดับน้ำ ผลการดำเนินงาน การทดลองการทำงานของระบบสามารถทำงานได้ดี การทดลองปลูกผักสลัดกรีนโอ๊คและредโอ๊ค ทำการทดลองทั้งหมด 42 วัน

บุญญฤทธิ์ วงศ์และคณะ (2016) ประยุกต์ใช้พลังงานไฟฟ้าจากแสงอาทิตย์ในโรงเรียนไฮโดรโปนิกส์ของกลุ่มวิชาหกิจชุมชนผักปลอดภัย งานวิจัยนี้ได้ลงพื้นที่พบกลุ่มวิชาหกิจปลูกผักไฮโดรโปนิกส์ ต. เกาะตาเลียง อ.ศรีสำโรง จ.สุโขทัย พบปัญหาหลายด้าน หนึ่งในปัญหาของกลุ่มวิชาหกิจนั้นคือ ต้องการลดใช้พลังงานไฟฟ้าสำหรับในโรงเรียนปลูกผักและวิจัยจึงออกแบบระบบการพัฒนาโรงเรียนปลูกผักระบบอัจฉริยะพัฒนาไฟฟ้าจากแสงอาทิตย์เพื่อลดต้นทุนการผลิต ผลประหยัดที่จากการใช้พลังงานไฟฟ้าร่วมกับระบบผลิตไฟฟ้าจากแสงอาทิตย์แบบอัตโนมัติ ใน 1 เดือน สามารถประหยัดได้ 77.2 หน่วย โดยเฉลี่ย 2.57 หน่วยต่อวัน คิดเป็นผลประหยัดค่าไฟฟ้า 386 บาทต่อเดือน (คิดหน่วยละ 5 บาท) คิดเป็นผลประหยัด 51.41% ระยะเวลาคืนทุน 7.88 ปีการวิจัยสามารถนำผลการวิจัยไปใช้ได้จริงและลดค่าไฟฟ้าแบบเดิมลงได้

อุษาวาดี ตันติราณุรักษ์ (2020) ศึกษาอัตราการระบายอากาศธรรมชาติของโรงเรียนเกษตรกรรมน้ำใช้ในการทำนายอัตราการระบายอากาศในโรงเรียนเกษตรกรรมที่มีช่องเปิดด้านบนและด้านข้าง ภายในใต้เงื่อนไขสภาพอากาศภายนอก ได้แก่ อุณหภูมิอากาศแวดล้อมและรังสีอาทิตย์ภายนอก โดยอัตราการ

ระยะอากาศคำนวณจากแรงดึงดูดตัวเนื่องมาจากการร้อนและแรงลม ความถูกต้องของสมการที่คำนวณได้จากแบบจำลองทางคณิตศาสตร์ที่ได้จากการสมดุลพลังงาน และการวัดอัตราความเร็วของอากาศ ผ่านพื้นที่ซึ่งเปิดด้านบนและด้านข้างให้ผลใกล้เคียงกันดี และสามารถนำไปใช้ทำนายอุณหภูมิอากาศภายในโรงเรือนเกษตรกรรมที่มีช่องเปิดด้านบนและ ด้านข้างได้

olgongt กองมนีและคณะ (2020) พัฒนาระบบอัจฉริยะต้นแบบให้เกษตรกร ซึ่งเป็นการพัฒนาเทคโนโลยีการให้ปุ๋ยในระบบนำ้ด้วยอินเทอร์เน็ตของสรรพสิ่ง เทคโนโลยี เชิฟเวอร์ และเทคโนโลยีอุปกรณ์เคลื่อนที่ ระบบดันแบบมีศักยภาพที่จะเปลี่ยนขั้นตอนที่ผู้ปลูกผักสดให้ในปัจจุบัน งานวิจัยใช้ระบบนี้เพื่อให้การใช้น้ำ ใช้ปุ๋ยน้ำมาก ใช้สารชีวภัณฑ์ในการผลิตผลักลัตอินทรีย์ในโรงเรือน ในช่วงฤดูหนาวและฤดูร้อน พ.ศ. 2561 ให้เหมาะสมที่สุด โรงเรือนตั้งอยู่ที่สำนักฟาร์ม มหาวิทยาลัยแม่โจ้ ผลการทดลองแสดงให้เห็นว่าระบบที่นำเสนอทำงานอย่างมีประสิทธิภาพ และเหมาะสมสำหรับการปลูกผักสดอินทรีย์ แต่ก็มีข้อผิดพลาดและบกพร่องเล็กน้อยซึ่งประกอบด้วย ความยากในการดูแลระบบ และความสามารถในการปรับระบบเมื่อมีปุ๋ยน้ำมากที่ใช้มีปริมาณธาตุอาหารไม่สม่ำเสมอหรือเป็นมาตรฐาน

Ihham Ihoume et al. (2022) ศึกษาให้ข้อมูลเชิงลึกใหม่เกี่ยวกับศักยภาพของโรงเรือนเพื่อเริ่มปรับปรุงคุณภาพและเพิ่มผลผลิตของสตอร์เบอร์รี่ โดยประสิทธิภาพการระบายความร้อนของระบบนี้แปลงโดยการปรับปรุงสภาพภูมิอากาศ การเพิ่มขั้นของอุณหภูมิอากาศภายในของเรือนจะจาก 5 °C และ 9 °C และการลดลงของความชื้นสัมพัทธ์ 24% และ 32% เมื่อเทียบกับเรือนกระจกควบคุมและสภาพแวดล้อมกลางแจ้งตามลำดับ มันปรับปรุงสภาพการเจริญเติบโตของพืชสตอร์เบอร์รี่ซึ่งส่งผลให้การเก็บเกี่ยวเร็วขึ้น 17 วัน และเพิ่มผลผลิตสตอร์เบอร์รี่ทั้งหมด 29% ในขณะที่ยังลดการใช้น้ำลง 30% ในช่วงฤดูหนาว การใช้ Internet of Things ของระบบทำให้สามารถตรวจสอบและวิเคราะห์ข้อมูลแบบเรียลไทม์จากสถานที่ได้กีด้วยทางเว็บไซต์ จากผลการศึกษาทางเศรษฐกิจพบว่าระบบนี้มีประโยชน์ในแง่ของการลงทุนและการประหยัดน้ำและพลังงาน

Hamza Benyenza et al. (2023) การควบคุมสภาพอากาศเรือนกระจกและการจัดการชลประทานเป็นความท้าทายที่สำคัญในการเกษตร การทำให้เป็นติดจิทัลของการเกษตรมีพัลส์ในการปฏิวัติ และเชื่อมกับความท้าทายโดยการปรับปรุงผลผลิตและผลกำไรอย่างมีนัยสำคัญที่ความนี้นำเสนอแพลตฟอร์มที่ใช้ IoT สำหรับการตรวจสอบและควบคุมสภาพแวดล้อมทางการเกษตร. แพลตฟอร์มนี้ช่วยให้สามารถควบคุมการชลประทานและสภาพอากาศภายในผ่านระบบพิวชั่นตามกลยุทธ์ตระกระท์คูล์เครือ นอกจากนี้ยังรวมพารามิเตอร์ทางการเกษตรโดยใช้ WSN ต้นทุนต่ำและจัดการและจัดเก็บโดยใช้ฐานข้อมูล MySQL ผลการทดสอบแสดงให้เห็นว่าประสิทธิภาพของระบบพิวชั่นช่วยให้สามารถสร้างตัวแปรในสภาพการเจริญเติบโตที่ดีที่สุดสำหรับพืชโดยใช้น้ำและพลังงานน้อยที่สุด ข้อต่อของพารามิเตอร์ภายนอกเรือนจะมีอิทธิพลต่อการตรวจสอบเรือนกระจกจากระยะไกลและลดความลำบากของเกษตรกรแล้ว HMI ที่พัฒนาขึ้นยังมีข้อดีหลายประการสำหรับผู้ใช้งานน้อยที่สุด ข้อต่อของพารามิเตอร์ภายนอกเรือนจะมีอิทธิพลต่อการตรวจสอบเรือนกระจกจากระยะไกลและลดความลำบากของเกษตรกรแล้ว

Kenny Kueh Yung Shin et al. (2022) ทำการศึกษาด้านไอโอตีโอเพนิกส์ IOT ชุดใหม่เชิงได้รับการขานนามว่า SMART GROW ใช้เพื่อตรวจสอบและควบคุมแม่むนต่าง ๆ ของระบบตามพารามิเตอร์พื้นฐานที่สำคัญในการปลูกพืชที่มีสุขภาพดี โดยความท้าทายที่ต้องเผชิญระหว่างการสร้างนี้คือการอ่านเซ็นเซอร์อนาคตที่ผิดปกติเมื่อเชื่อมต่อกับไมโครคอนโทรลเลอร์บอร์ดเดียว (ESP32) ปัญหานี้ได้รับการแก้ไขแล้ว ปัจจุบัน SMART GROW สามารถตรวจสอบพารามิเตอร์พื้นฐาน เช่น pH, EC และระดับน้ำ และสามารถรองรับเซ็นเซอร์เพิ่มเติมสำหรับการตรวจสอบพารามิเตอร์อื่น ๆ หากจำเป็น SMART GROW สามารถทำซ้ำและสร้างที่บ้านได้อย่างง่ายดายและปรับแต่งตามความต้องการของโรงงาน SMART GROW มีความหลากหลายเนื่องจากสามารถใช้ปลูกพืชได้หลากหลายรวมถึงสมุนไพรผักและผลไม้และมีประโยชน์หลายประการเหนือกว่าการปลูกบนดินแบบดั้งเดิม เช่น การควบคุมระดับน้ำโดยอัตโนมัติ

Kannan M et al. (2023) ได้สังเกตประโยชน์และข้อ จำกัด ที่แตกต่างกันของระบบการปลูกพืชไร้ดินผ่านการเพาะปลูกพืชต่าง ๆ เช่นผักชีและผักใบ ใช้เวลาน้อยลงสำหรับผลผลิตและไม่มีโรคที่เกิดจากดิน การจัดการพืชที่ง่ายการเจริญเติบโตของวัชพืชเป็นศูนย์และ ไม่มีข้อกำหนดในการฉีดพ่นน้ำ ซึ่งเป็นหนึ่งในข้อดีบางประการของระบบการปลูกพืชไร้ดิน ระบบการปลูกพืชไร้ดินสามารถรับประกันการประหยัดน้ำขั้นต่ำได้ถึง 70 – 80% เมื่อเทียบกับระบบการทำฟาร์มบนดิน และอธิบายถึงความสำคัญข้อดีและข้อเสียของการการทำฟาร์มไฮดร็อปเปนิกส์

Parastoo Mohebi et al. (2023) ทำการศึกษาเรื่องการเพิ่มประสิทธิภาพสำหรับการออกแบบร่วมกันและการดำเนินงานรายชั่วโมงของระบบพลังงานแสงอาทิตย์ให้ความร้อนเรือนกระจก สำหรับระบบพลังงานไฮบริดประกอบด้วยตัวสะสมพลังงานแสงอาทิตย์การจัดเก็บความร้อนในระยะยาวและระยะสั้นและหม้อไอน้ำสำรอง โซลูชันต้นทุนขั้นต่ำสามารถเข้าถึงได้โดยใช้ตัวแก้ Cplex หลังจากนั้นผลลัพธ์ที่ดีที่สุดจะประสบความสำเร็จโดยการพิจารณาอย่างการลดควรบอนการจัดตารางการเพาะปลูกเรื่องผลกระทบจากการกำหนดราคาก๊าซธรรมชาติและสถานการณ์ต้นทุนการลงทุนของตัวเก็บพลังงาน แสงอาทิตย์ ในที่สุดการใช้วิธีการ epsilon-constraint การเพิ่มประสิทธิภาพหลายวัตถุประสงค์จะดำเนินการด้วยการลด CO₂ การปล่อยมลพิษและค่าใช้จ่ายรายปีตามวัตถุประสงค์ LINMAP กำหนดโซลูชันที่เหมาะสมที่สุดของด้านหน้า Pareto ไปยังจุดที่เหมาะสมที่สุดและขนาดและการไฟลของพลังงานที่เหมาะสม

บทที่ 3

ขั้นตอนและวิธีดำเนินการ

3.1 ขั้นตอนการดำเนินการ

ขั้นตอนและวิธีการดำเนินงานจะเป็นการวางแผนในส่วนการดำเนินการโดยจะมีการศึกษาเครื่องมือวัดระบบต่างๆ ภายในตัวโรงเรือน เช่นระบบควบคุมอุณหภูมิ ระบบแสง อีกทั้งยังศึกษาปัจจัยที่มีผลต่อพืช การสำรวจ และรวบรวมข้อมูล ณ ศูนย์วิจัยเกษตรสมัยใหม่ ฟาร์ม 900 ไร่ มหาวิทยาลัยแม่โจ้ เพื่อนำมาออกแบบการแสงผลข้อมูลของโรงเรือน ให้มีประสิทธิภาพมากขึ้น โดยมีขั้นตอนดังต่อไปนี้



รูปที่ 3.1 แผนภาพขั้นตอนการดำเนินงาน

3.1.1 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการออกแบบระบบตรวจวัดโรงเรือนโดยใช้ระบบ IOT อันประกอบไปด้วย ทฤษฎีของโรงเรือนทั้งในส่วนของประเภท วัสดุ ปัจจัยที่ส่งผลต่อโรงเรือน ระบบตรวจวัดโรงเรือน และมาตรการณ์ของแผนกรอนุรักษ์พลังงาน 2015

3.1.2 สำรวจอุปกรณ์และจุดตรวจภายในโรงเรือน

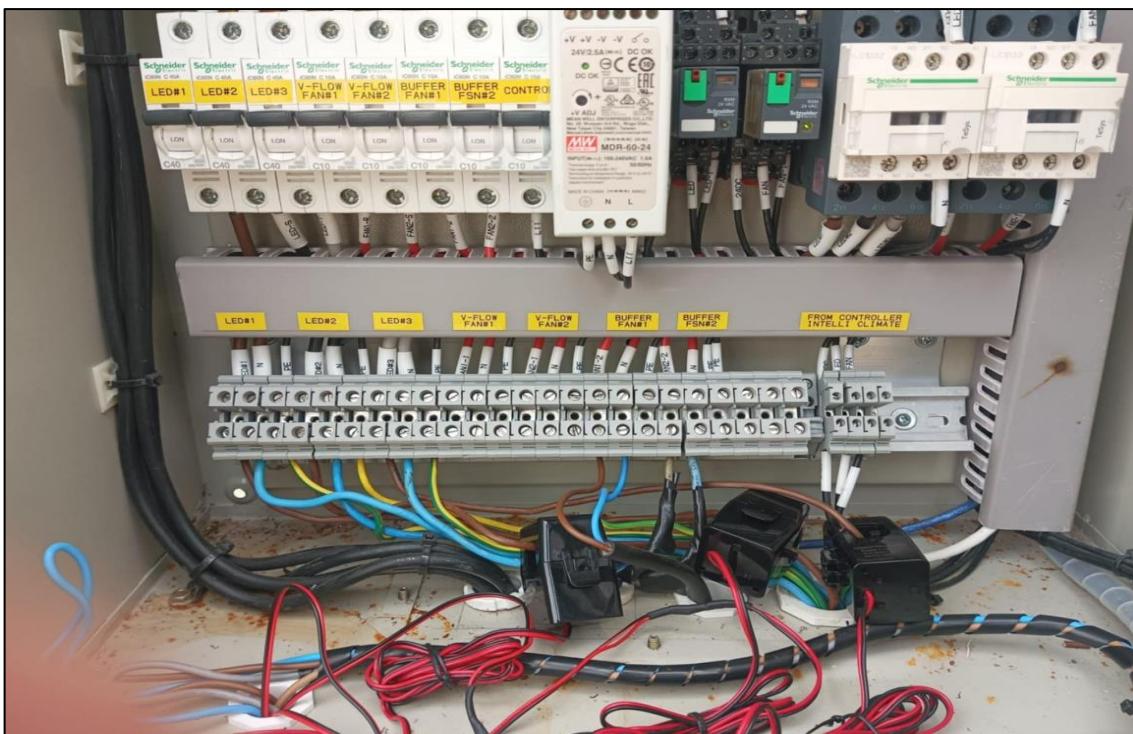
ส่วนนี้จะเป็นการสำรวจเพื่อหาตำแหน่งอุปกรณ์ ระบบแสง ระบบระบายอากาศ ระบบทำความเย็นแบบนะเหยย ระบบปั๊มน้ำและถังปั๊ย ระบบตรวจวัด และระบบควบคุมที่ติดตั้งในโรงเรือนเกษตร และการจัดวางของพืชภายในโรงเรือน



รูปที่ 3.2 สำรวจอุปกรณ์และจุดตรวจภายในโรงเรือน 1



รูปที่ 3.3 MCB Climate controller



รูปที่ 3.4 ระบบแสงสว่าง



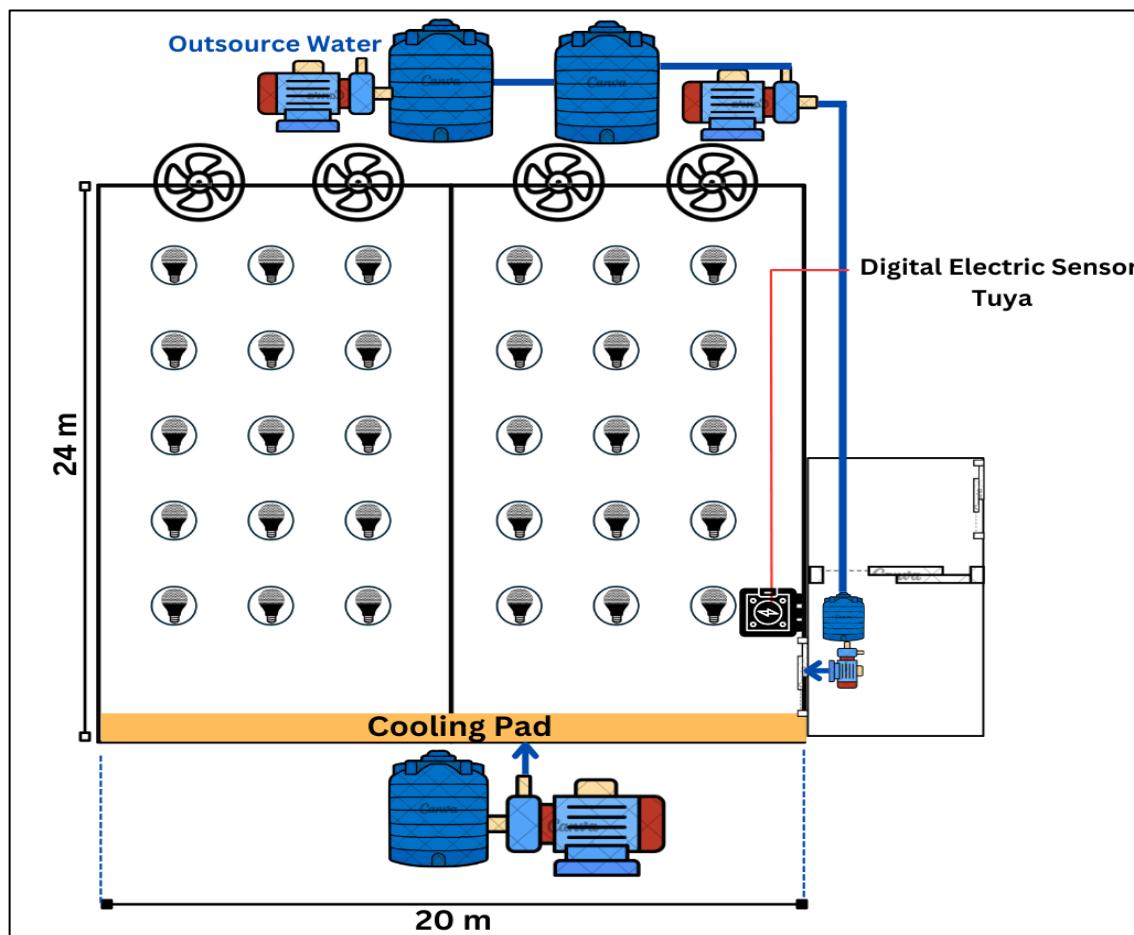
รูปที่ 3.5 Outdoor Weather Station

หลังจากตรวจสอบเซนเซอร์ พบว่า มีเซนเซอร์ วัดอุณหภูมิและความชื้นจำนวน 6 อัน เซนเซอร์วัดไฟฟ้า จำนวน 9 ชิ้น และเซนเซอร์วัดค่าความสว่าง จำนวน 2 ชิ้น

ชื่อ	จำนวน
Temperature electric quantity controller Tuya	6
Compteur Digital Electric Tuya	9
BH1750	2

3.1.3 จำลองแผนผังการวางแผนอุปกรณ์

จากการสำรวจและเก็บข้อมูลตำแหน่งของอุปกรณ์ต่างๆ ภายในโรงเรือนเกษตร ได้แก่ระบบแสงรวมถึงหลอดไฟ, โคมไฟ, และอุปกรณ์ควบคุมแสง ระบบระบายอากาศ พัดลม, ช่องระบายอากาศ, และระบบควบคุมความชื้น ระบบทำความสะอาด ระบบระบายความเย็น ระบบระบายความเย็นระบบให้น้ำและปุ๋ย ปั๊มน้ำ, ถังปุ๋ย, และท่อส่งน้ำ ระบบตรวจวัด เซนเซอร์วัดอุณหภูมิ, ความชื้น, แสง, ค่า EC, pH และอื่นๆ ระบบควบคุมคอมพิวเตอร์, ไมโครคอนโทรลเลอร์, และซอฟต์แวร์ควบคุมได้มีการสร้างแบบจำลองแสดงตำแหน่งและการเชื่อมต่อของอุปกรณ์ต่างๆ รวมถึงการจัดวางพื้นที่ในโรงเรือน เพื่อใช้เป็นฐานในการออกแบบจุดติดตั้งเซนเซอร์เพิ่มเติม สำหรับการตรวจวัดค่าต่างๆ ที่เกี่ยวข้องกับการเจริญเติบโตของพืช

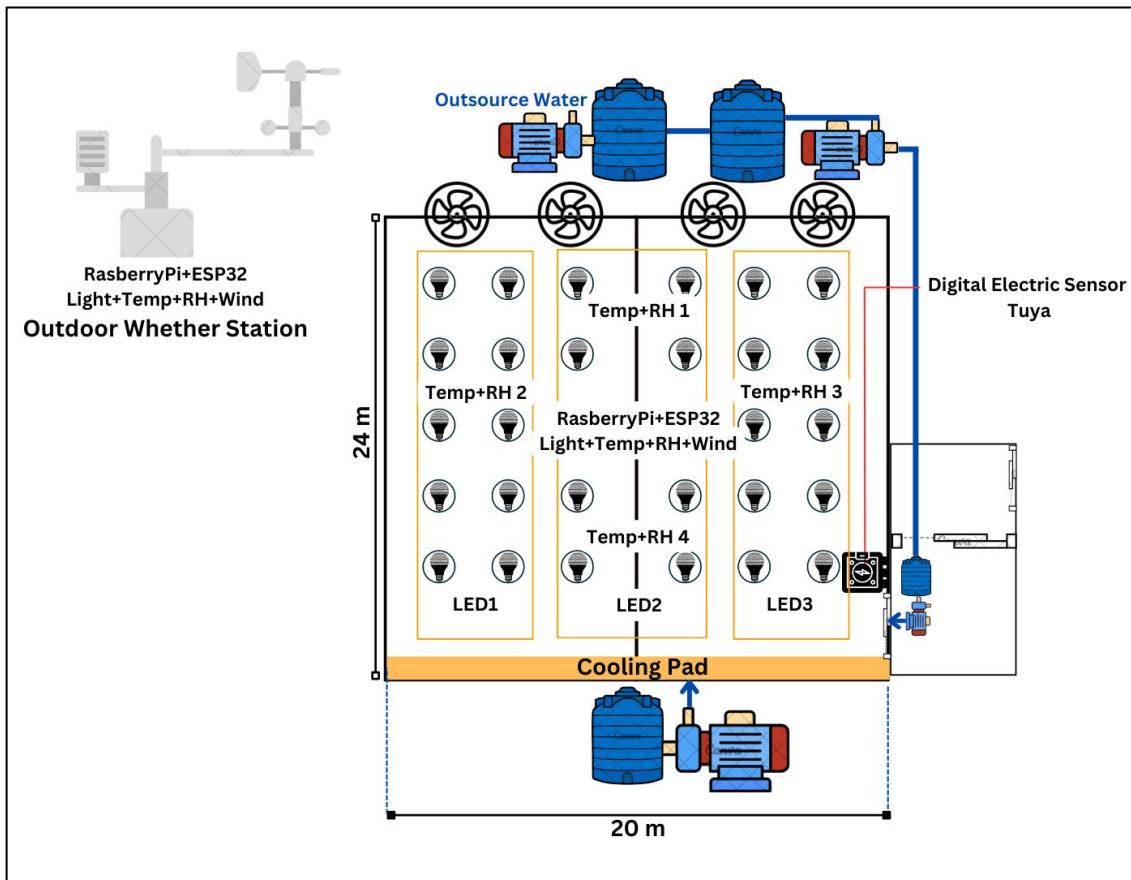


รูปที่ 3.6 แผนผังระบบแสงสว่างและระบบน้ำของโรงเรือนกัญชา

3.1.4 การออกแบบจุดติดตั้งเซ็นเซอร์หรือเลี้ยงเอาท์จุดตรวจวัดต่างๆ

จากแผนผังโรงเรือนในการออกแบบจุดติดตั้งเซ็นเซอร์จะเน้นที่การครอบคลุมพื้นที่ทั้งหมดของโรงเรือนและการวัดค่าที่สำคัญต่อการเรียบเติบโตของพืช โดยพิจารณาจากตำแหน่งของอุปกรณ์ต่างๆ เช่น แหล่งกำเนิดแสง พัดลมระบบอากาศ และระบบให้น้ำ เพื่อให้ได้ข้อมูลที่แม่นยำและครอบคลุมมากที่สุด และได้ออกแบบจุดติดตั้งเซ็นเซอร์ได้ดังนี้

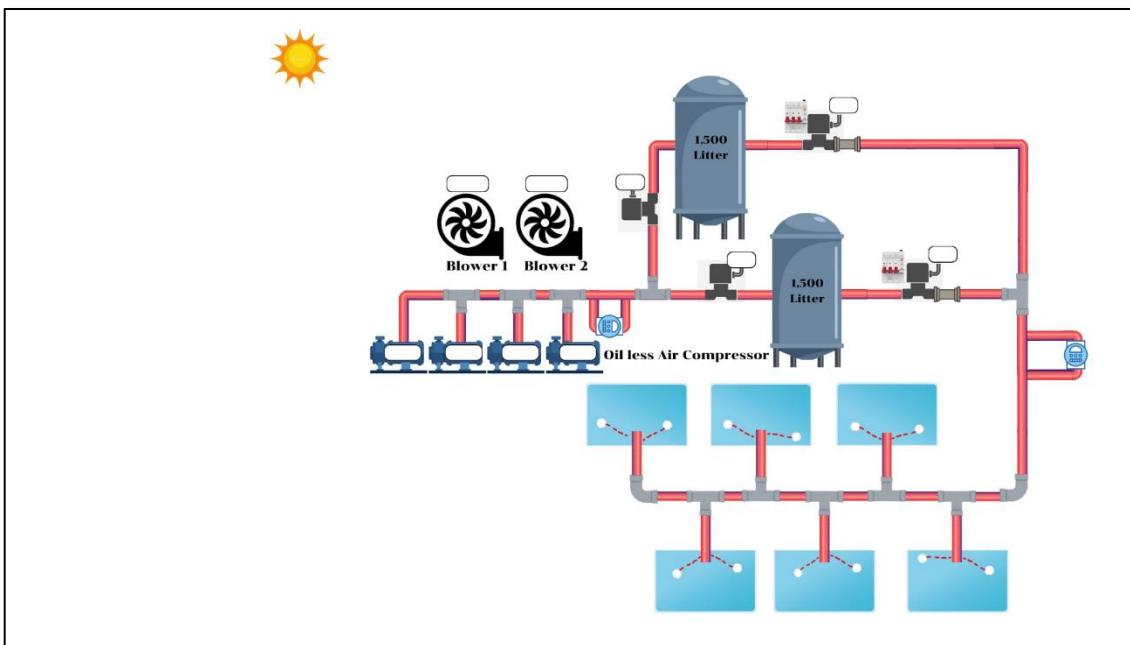
3.1.4.1 การออกแบบจุดติดตั้งเซ็นเซอร์หรือเลี้ยงเอาท์จุดตรวจวัดต่างๆ ในโรงเรือนกัญชา ณ ฟาร์ม 900 ไร่ มหาวิทยาลัยแม่โจ้



รูปที่ 3.7 แผนผังจุดติดตั้งเซ็นเซอร์ของโรงเรือนกัญชา

3.1.4.1 การออกแบบจุดติดตั้งเซ็นเซอร์หรือเลี้ยงเอาท์จุดตรวจวัดต่างๆ ในโรงเรือนบ่อเก็บน้ำ

การออกแบบระบบที่ใช้ในการจ่ายและควบคุมการไหลของน้ำและอากาศในบ่อเก็บน้ำ มีส่วนประกอบหลัก ๆ คือ ถังเก็บน้ำ (Tanks) มีถังเก็บน้ำขนาด 1,500 ลิตร จำนวนสองถัง ใช้ในการเก็บน้ำและจ่ายน้ำในระบบ มี Blower 2 ตัวที่ใช้ในการเป่าอากาศเข้าสู่ระบบห้องเพื่อตอบสนองต่อความต้องการใช้อากาศในระบบ Oil-less Air Compressor เครื่องยัดอากาศแบบไม่ใช่น้ำมันสำหรับการจ่ายอากาศไปยังท่อและถังเก็บต่าง ๆ ในระบบ ระบบห้อง (Piping System) ระบบห้องเชื่อมต่อถังเก็บน้ำ, เครื่องอัดอากาศ, และ Blower ทั้งหมดเข้าด้วยกัน ท่อมีวาร์ล์และจุดเชื่อมต่าง ๆ ที่ช่วยในการควบคุมทิศทางการไหลของน้ำและอากาศในระบบ ถังน้ำ (Water Chambers) มีถังน้ำหลายถังเชื่อมต่อกับระบบห้อง ทำหน้าที่เป็นจุดที่น้ำจะถูกปล่อยหรือรับน้ำเข้ามาตามความจำเป็น



รูปที่ 3.8 แผนผังจุดติดตั้งเซนเซอร์ของบ่อกุ้ง

3.1.5 ติดตั้ง Home assistant server

ในการติดตั้ง Home Assistant บน Raspberry Pi มีอุปกรณ์ คือ 1.) Raspberry Pi รุ่นที่รองรับ (แนะนำให้ใช้ Raspberry Pi 4 ขึ้นไป) 2.) การ์ด microSD ที่มีความจุอย่างน้อย 32 GB และมีคุณภาพดี 3.) คอมพิวเตอร์ที่เชื่อมต่ออินเทอร์เน็ต เพื่อทำการดาวน์โหลดและติดตั้งการเซ็ตอัปกับอินเทอร์เน็ต สำหรับ Raspberry Pi (ผ่านสาย LAN หรือ Wi-Fi) 4.) อะแดปเตอร์จ่ายไฟที่เพียงพอต่อการใช้งาน Raspberry Pi

- 1.) ดาวน์โหลด Raspberry Pi Imager สำหรับ Windows กดปุ่ม Download for Windows เพื่อดownloadโปรแกรม Raspberry Pi Imager สำหรับระบบปฏิบัติการ Windows สำหรับ macOS หรือ Ubuntu กดลิงก์ที่ระบุไว้ใต้ปุ่มดาวน์โหลด Windows เพื่อเลือกดาวน์โหลดสำหรับ macOS หรือ Ubuntu ตามความเหมาะสมของระบบเมื่อดาวน์โหลดเสร็จ ให้เปิดไฟล์ที่ดาวน์โหลดมาเพื่อติดตั้ง Raspberry Pi Imager บนคอมพิวเตอร์ นำ MicroSD card ที่จะใช้กับ Raspberry Pi เสียบเข้ากับ คอมพิวเตอร์ผ่านเครื่องอ่านการ์ด (SD card reader)

Raspberry Pi OS

Your Raspberry Pi needs an operating system to work. This is it. Raspberry Pi OS (previously called Raspbian) is our official supported operating system.

Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

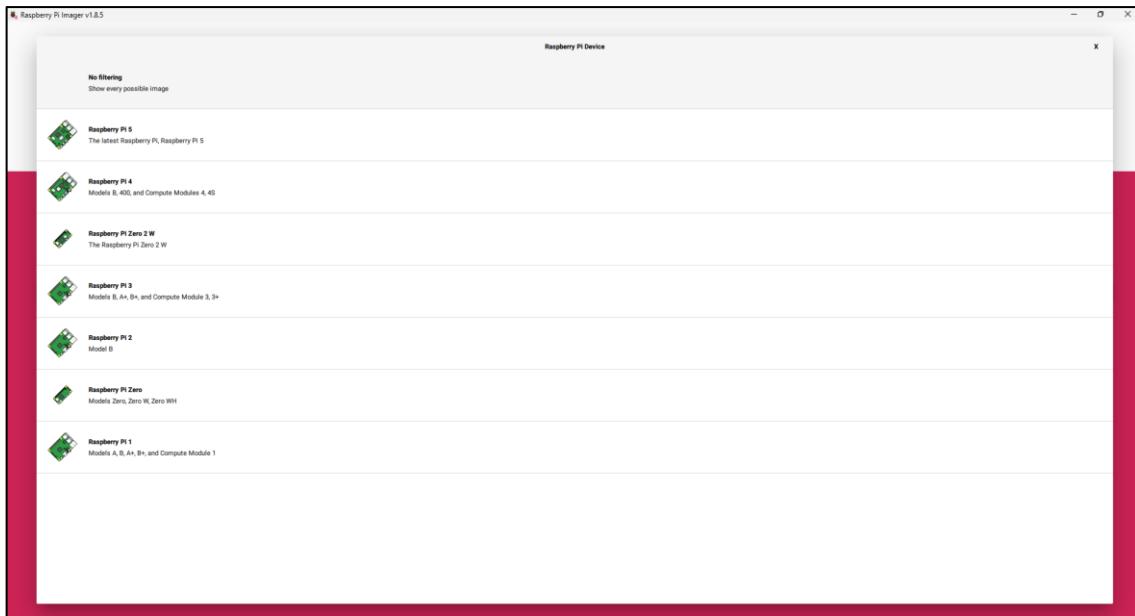
[Download for macOS](#)

[Download for Ubuntu for x86](#)

To install on **Raspberry Pi OS**, type
`sudo apt install rpi-imager`
in a Terminal window.

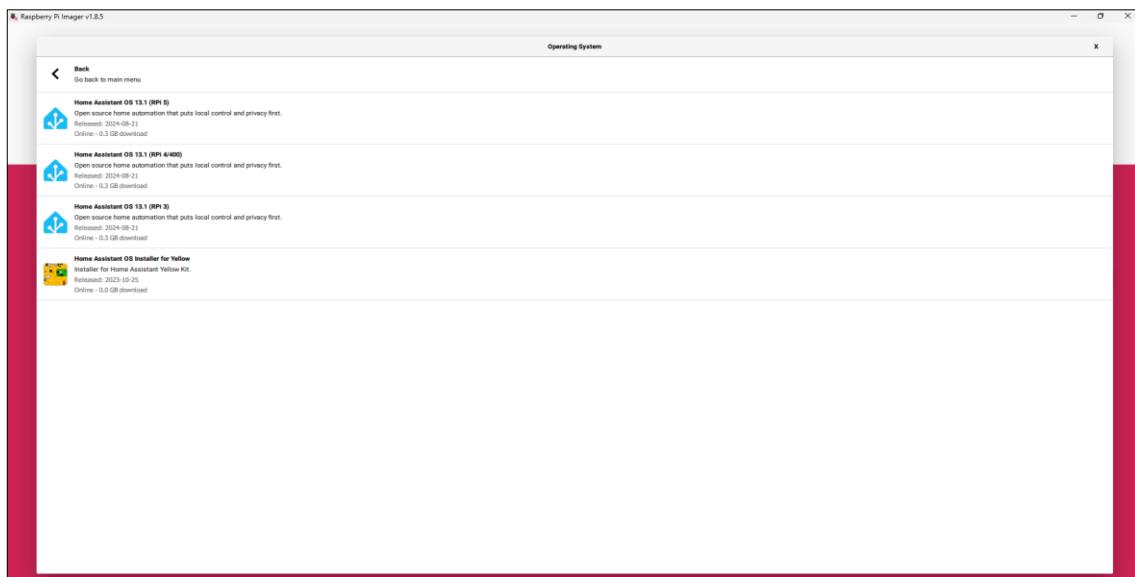
รูปที่ 3.9 หน้าติดตั้ง Raspberry Pi imager

2.) เลือกบอร์ด Raspberry Pi ในรายการที่ปรากฏบนหน้าจอ เลือกบอร์ดที่กำลังใช้อยู่ โดยในภาพมีหลายรุ่นให้เลือก เช่น Raspberry Pi 5 สำหรับบอร์ด Raspberry Pi 5, Compute Module 4S Raspberry Pi 4 สำหรับบอร์ด Raspberry Pi 4, Compute Module 4, 4S Raspberry Pi Zero 2 W สำหรับ Raspberry Pi Zero 2 W Raspberry Pi 3 สำหรับบอร์ด Raspberry Pi 3, Compute Module 3+ Raspberry Pi 2 สำหรับ Raspberry Pi 2 และรุ่น Zero W Raspberry Pi 1 สำหรับบอร์ดรุ่นเก่าของ Raspberry Pi เช่น รุ่น A, B และ Compute Module 1 โดยการคลิกเลือกบอร์ดที่ตรงกับรุ่น Raspberry Pi โปรแกรมจะกรองและแสดงภาพระบบปฏิบัติการที่รองรับตามรุ่นที่เลือก



รูปที่ 3.10 เลือกบอร์ด Raspberry Pi

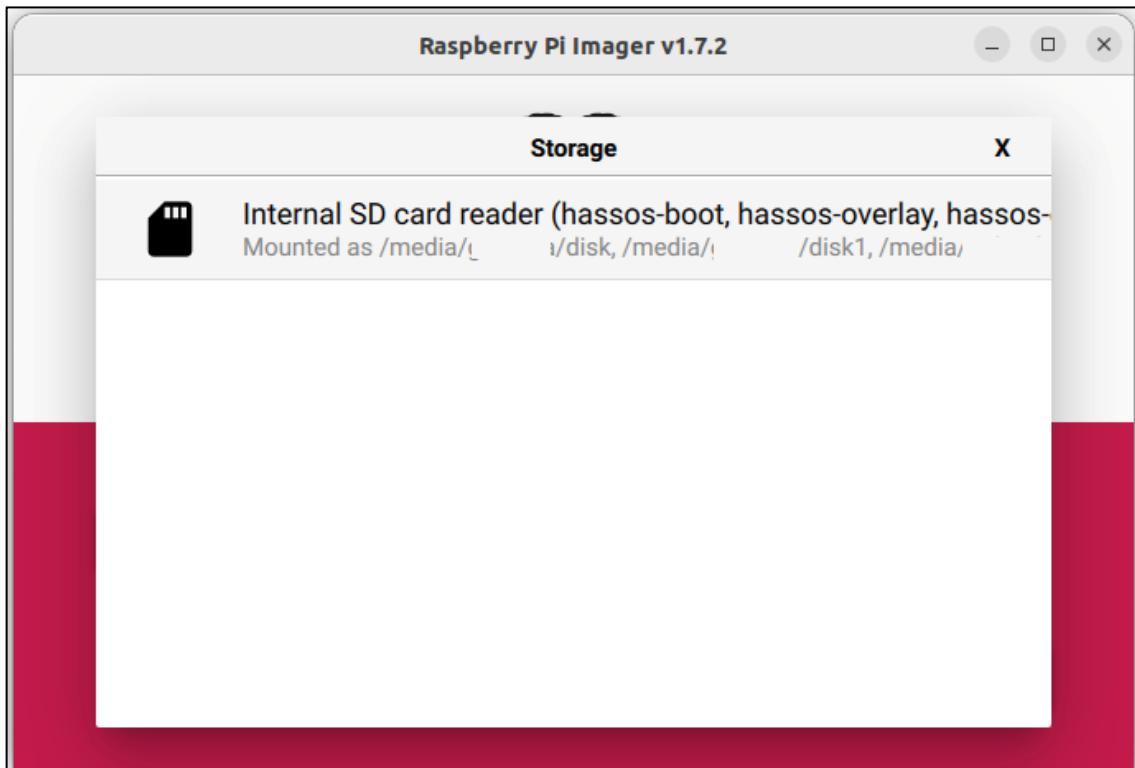
3.) เลือก Operating System (OS) สำหรับติดตั้งใน Raspberry Pi ซึ่งในภาพนี้มีตัวเลือกระบบปฏิบัติการ Home Assistant OS หลายเวอร์ชันให้เลือก โดยมีขั้นตอนดังนี้ เลือกระบบปฏิบัติการที่ต้องการ ในหน้าจอที่ปรากฏ จะมีหลายตัวเลือกของ Home Assistant OS Home Assistant OS 13.5 (RPI) สำหรับ Raspberry Pi ที่รองรับ โดยมีขนาดไฟล์ประมาณ 0.2 GB Home Assistant OS 13.5 (RPI 4/400) สำหรับ Raspberry Pi รุ่น 4 และ 400 ขนาดไฟล์ประมาณ 0.2 GB Home Assistant OS 13.5 (RPI 3) สำหรับ Raspberry Pi รุ่น 3 ขนาดไฟล์ 0.2 GB Home Assistant OS Installer for Yellow สำหรับ Home Assistant Yellow ขนาดไฟล์ประมาณ 0.2 GB เลือกเวอร์ชันที่ตรงกับบอร์ด Raspberry Pi ตามที่เลือกในขั้นตอนก่อนหน้า คลิกที่ตัวเลือกนั้น โปรแกรมจะทำการดาวน์โหลดไฟล์ระบบปฏิบัติการจากเซิร์ฟเวอร์



รูปที่ 3.11 การติดตั้ง Home assistant

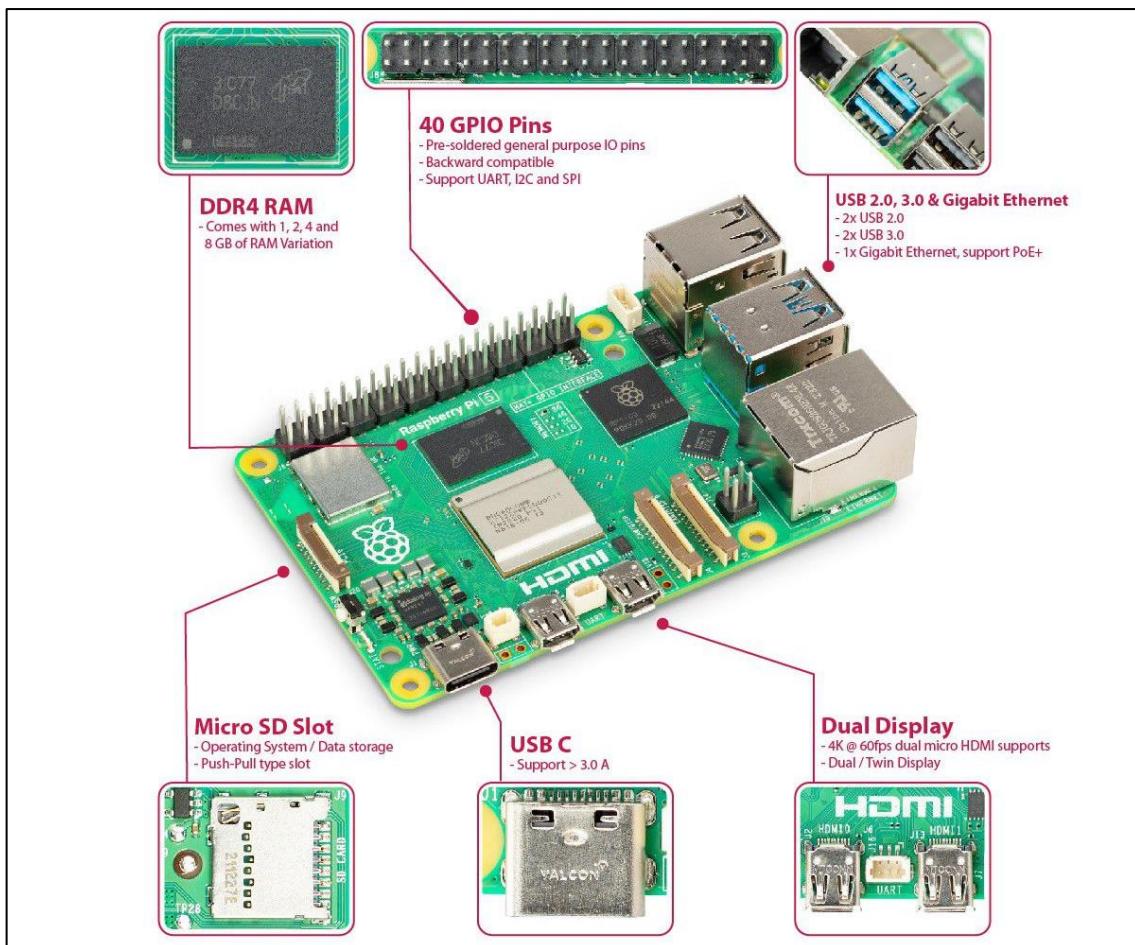
4.) หลังจากที่เลือกรอบปฎิบัติการและทำการดาวน์โหลดเสร็จสมบูรณ์แล้ว โปรแกรมจะดำเนินการเขียนไฟล์ระบบทั้งหมดลงใน MicroSD card ที่ได้เชื่อมต่อไว้กับคอมพิวเตอร์ ขั้นตอนนี้สำคัญมาก เนื่องจาก MicroSD card จะเก็บข้อมูลสำคัญทั้งหมดที่ Raspberry Pi จำเป็นต้องใช้ในการบูต Home Assistant OS อย่างถูกต้อง

หลังจากการเขียนไฟล์เสร็จสิ้นสามารถนำ MicroSD card นำไปใส่ในช่องใส่ของ Raspberry Pi ได้ทันที เมื่อติดตั้ง MicroSD card แล้วสามารถเริ่มบูตระบบ Raspberry Pi ได้โดยการเสียบปลั๊กไฟ ระบบจะเริ่มต้นกระบวนการบูตโดยอัตโนมัติ และ Home Assistant OS จะเริ่มทำงานอัตโนมัติหลังจากที่เสียบ MicroSD card

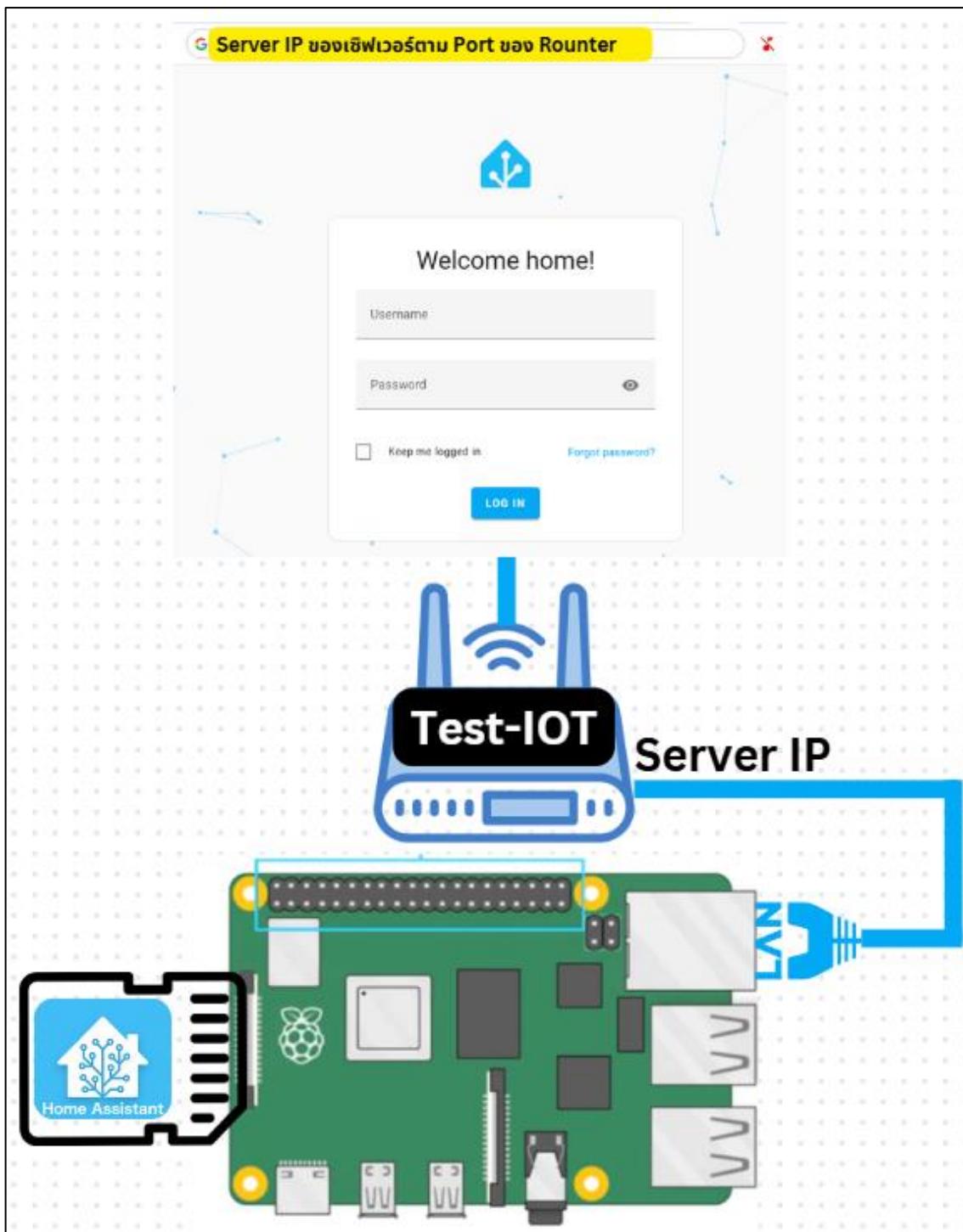


รูปที่ 3.12 ขั้นตอนการเลือก SD Card

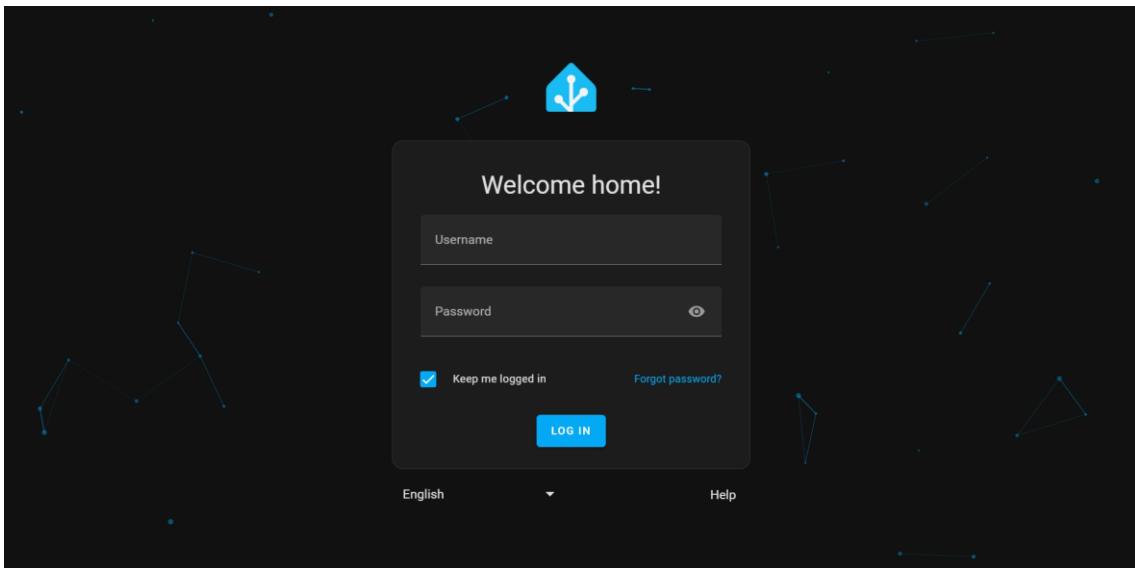
5.) ภายใต้ไม่กี่นาทีหลังจากเชื่อมต่อ Raspberry Pi จะสามารถเข้าถึง Home Assistant ได้ในเว็บเบราว์เซอร์ของระบบเดสก์ท็อป ให้พิมพ์ homeassistant.local:8123 หมายเหตุว่า ถ้าใช้ Windows เวอร์ชันเก่าหรือมีการตั้งค่าเครือข่ายที่เข้มงวดมากขึ้น จึงต้องเข้าถึง Home Assistant โดยใช้ homeassistant:8123 หรือ http://X.X.X.X:8123 (ให้แทน X.X.X.X ด้วย IP address ของ Raspberry Pi ของ) เวลาที่ใช้สำหรับหน้าเว็บนี้ขึ้นอยู่กับอาร์ดแวร์ที่ใช้ สำหรับ Raspberry Pi 4 หรือ 5 หน้าจอควรจะปรากฏภายใน 1 นาที ถ้าหลังจาก 5 นาทีแล้วยังไม่ปรากฏบน Pi 4 หรือ 5 อาจเกิดจากการเขียนอิมเมจไม่ถูกต้อง ลองแฟลช SD การ์ดใหม่ หรือเปลี่ยนการ์ด SD เป็นใบอื่น หากยังไม่สามารถแก้ไขได้ ให้ตรวจสอบผลลัพธ์จากคอนโซลของ Raspberry Pi ทำได้โดยเชื่อมต่อมอนิเตอร์ผ่าน HDMI



รูปที่ 3.13 การเชื่อมต่อของ Raspberry Pi5

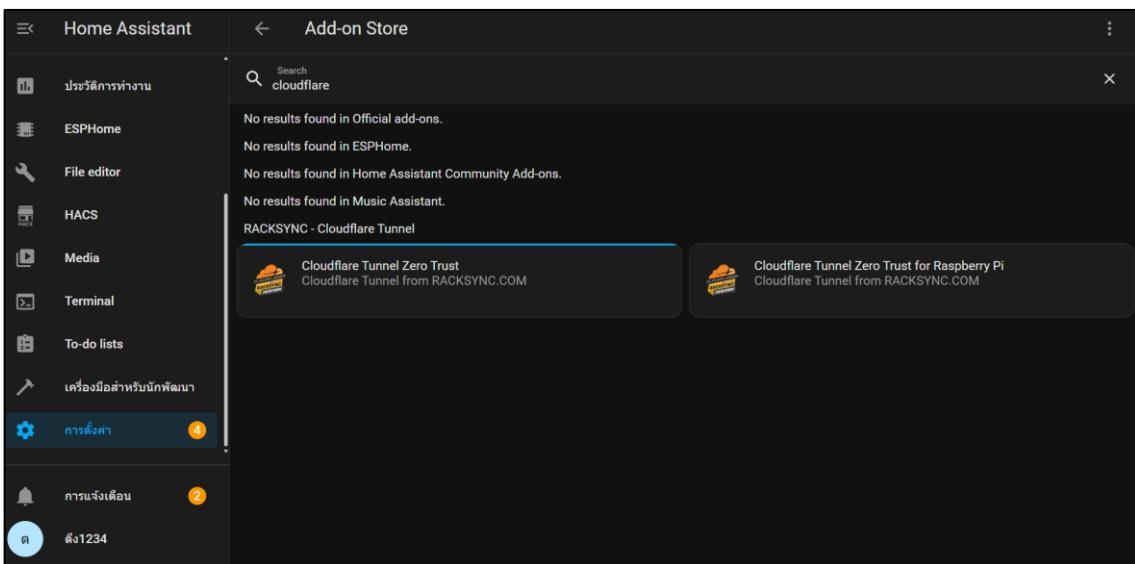


รูปที่ 3.14 การเชื่อมต่อให้ Raspberry Pi เป็นเซิฟเวอร์หลักของ Home assistant



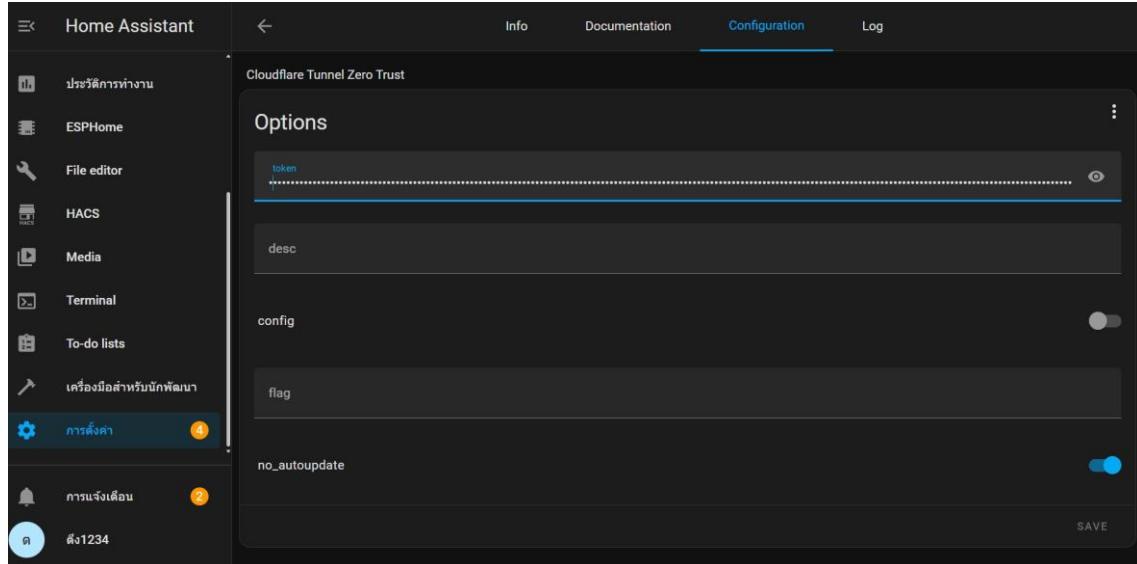
รูปที่ 3.15 หน้า Login ของ Home assistant

6.) การเข้าถึง Home Assistant จากระยะไกลอย่างปลอดภัย สามารถทำได้โดยใช้ Cloudflare Tunnel ซึ่งเป็นวิธีที่ช่วยให้สามารถเชื่อมต่อ Home Assistant กับอินเทอร์เน็ตได้โดยไม่จำเป็นต้องเปิดพอร์ตหรือทำการตั้งค่าเครือข่ายเพิ่มเติมที่บ้าน (port forwarding) โดย Cloudflare Tunnel จะสร้างการเชื่อมต่อที่ปลอดภัยผ่าน Cloudflare โดยตรงไปยัง Home Assistant ขั้นตอนการตั้งค่า Cloudflare Tunnel กับ Home Assistant เริ่มจากสร้างบัญชี Cloudflare หากยังไม่มีบัญชี Cloudflare จะต้องสมัครที่ Cloudflare เพื่อให้สามารถใช้บริการ Cloudflare Tunnel ได้ ติดตั้ง Cloudflare Tunnel บน Home Assistant สามารถติดตั้ง Cloudflare Tunnel ผ่าน Add-ons ใน Home Assistant ไปที่เมนู Supervisor จากนั้นเลือก Add-on Store ค้นหาและติดตั้ง Cloudflare add-on เมื่อติดตั้งเสร็จแล้ว ให้เปิดการตั้งค่า start on boot และ watchdog เพื่อให้แน่ใจว่า Tunnel จะทำงานเสมอเมื่อระบบบุต



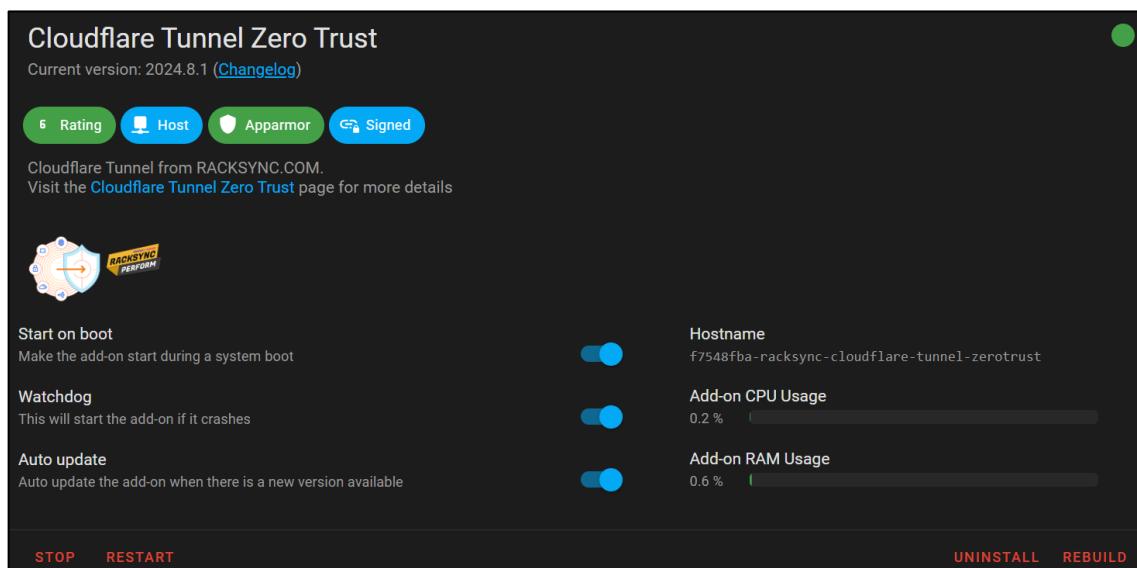
รูปที่ 3.16 เพิ่ม cloudflare add on

7.) หลังจากนั้นเปิดการตั้งค่าของ Cloudflare Add-on ใส่ข้อมูล API Token ที่คัดลอกมาจาก Cloudflare ในช่องที่ระบุ อาจต้องตั้งค่าเพิ่มเติมเกี่ยวกับโดเมนหรือ DNS ในขั้นตอนนี้เพื่อให้ Tunnel เชื่อมต่ออย่างถูกต้องบันทึกการตั้งค่าและเริ่มต้น Cloudflare Tunnel หลังจากใส่ Token แล้ว กดบันทึก และเริ่ม Add-on เพื่อทำให้ Tunnel เริ่มทำงาน

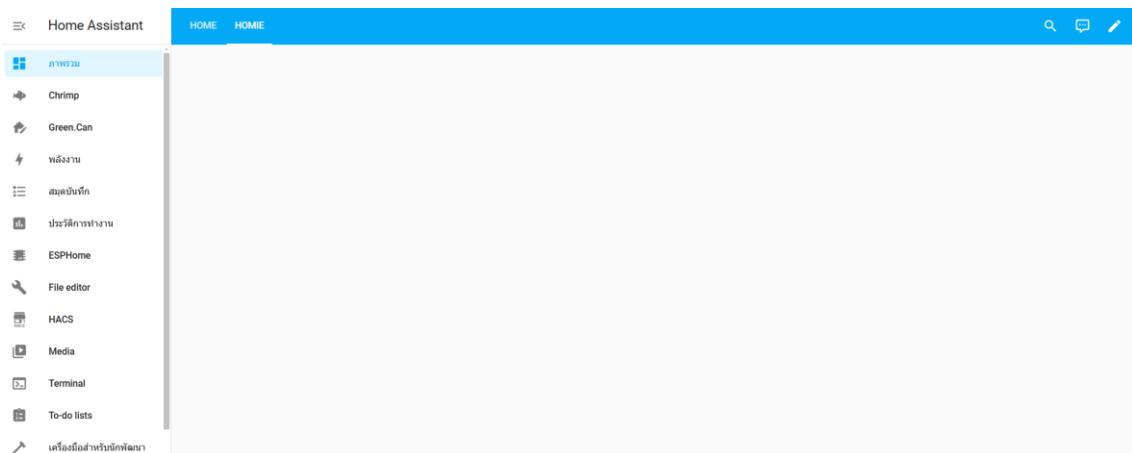


รูปที่ 3.17 Token ใน Cloudflare

8.) หลังจากที่ Tunnel ทำงานแล้ว จะต้องตั้งค่า DNS record เพื่อให้สามารถเข้าถึง Home Assistant ผ่านโดเมนที่ตั้งค่าไว้ใน Cloudflare (ตัวอย่างเช่น homeassistant.mydomain.com) ไปที่แดชบอร์ด Cloudflare และเพิ่ม DNS entry ของโดเมน โดยซื้อไปยัง Tunnel ที่สร้างไว้การตรวจสอบการเข้าถึงจากระยะไกล เมื่อการตั้งค่าเสร็จสมบูรณ์แล้ว สามารถเข้าถึง Home Assistant ได้จากระยะไกล โดยผ่าน URL ที่กำหนดไว้ใน Cloudflare จะช่วยปกป้องและเข้ารหัสการเชื่อมต่อทั้งหมด ทำให้สามารถเข้าถึงระบบได้อย่างปลอดภัย



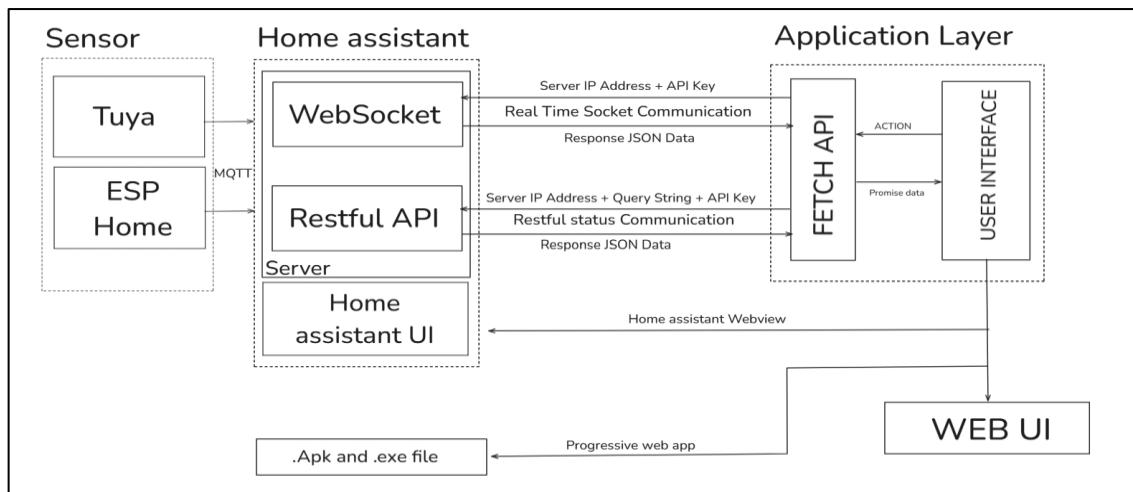
รูปที่ 3.18 Cloud flare tunnel



รูปที่ 3.19 หน้าจอแสดงผลพร้อมใช้งาน

3.1.6 หลักการทำงานของหน้าแสดงผล

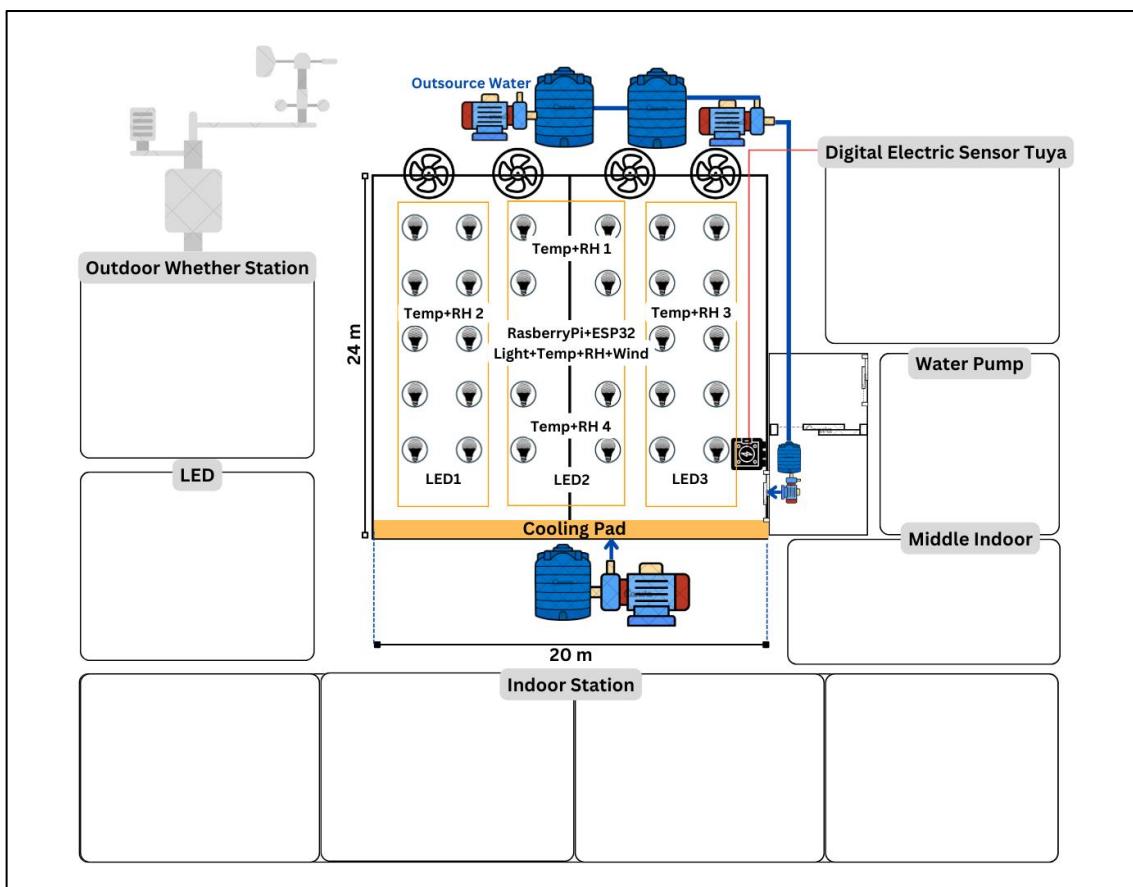
การสื่อสารและโครงสร้างการทำงานระหว่าง Home Assistant และ Application Layer ที่แสดงถึงวิธีการที่ข้อมูลถูกจัดการและแสดงผลผ่าน User Interface หรือหน้าแสดงผล UI ทำหน้าที่รับคำสั่งจากผู้ใช้และส่งคำสั่งนั้นกลับไปที่ระบบผ่าน Fetch API โดยผู้ใช้สามารถโต้ตอบกับแอปพลิเคชันผ่านหน้าแสดงผลนี้ คือ ดึงข้อมูลสถานะต่าง ๆ จากระบบ UI ยังได้รับ Promise data ซึ่งเป็นข้อมูลที่ถูกส่งกลับจากระบบทหลังจากที่ทำงานเสร็จสิ้น เช่น ข้อมูล JSON ที่ได้รับหลังจากการร้องขอข้อมูลจากเซิร์ฟเวอร์ Fetch API ใช้ในการสื่อสารกับ WebSocket หรือ RESTful API ในฝั่งเซิร์ฟเวอร์ ขึ้นอยู่กับลักษณะของคำขอ มีการส่งคำสั่งจากผู้ใช้ (ACTION) ไปยัง Fetch API ซึ่งจะทำการดึงข้อมูลจากเซิร์ฟเวอร์ ไม่ว่าจะเป็นการสื่อสารแบบเรียลไทม์ผ่าน WebSocket หรือการร้องขอข้อมูลสถานะผ่าน RESTful API WebSocket เป็นช่องทางการสื่อสารแบบเรียลไทม์ โดย Fetch API จะส่งคำขอพร้อม Server IP Address และ API Key เพื่อเชื่อมต่อกับ WebSocket บนเซิร์ฟเวอร์ เมื่อเชื่อมต่อแล้ว ข้อมูลในรูปแบบ JSON Data จะถูกส่งกลับมาที่ UI ผ่าน Fetch API เพื่อแสดงผลให้ผู้ใช้เห็นเหมาะสมสำหรับสถานการณ์ที่ต้องการการอัปเดตข้อมูลอย่างต่อเนื่อง เช่น การแสดงผลสถานะการทำงานของอุปกรณ์อัจฉริยะแบบเรียลไทม์ ในกรณีที่ต้องการดึงข้อมูลสถานะหรือการอัปเดตข้อมูลที่ไม่จำเป็นต้องเป็นแบบเรียลไทม์ การสื่อสารจะผ่าน RESTful API Fetch API จะส่งคำขอไปที่ RESTful API โดยระบุ Server IP Address, Query String, และ API Key ข้อมูลจะถูกส่งกลับมาในรูปแบบ JSON Data ผ่าน Fetch API ไปยังหน้าแสดงผล (UI) Web UI คือหน้าแสดงผลที่ผู้ใช้สามารถเข้าถึงได้ผ่านเบราว์เซอร์ โดยการทำงานทั้งหมดเกิดขึ้นภายใน Home Assistant Webview ที่ถูกฝังไว้ในแอปพลิเคชัน การใช้งานสามารถเป็นแบบ Progressive Web App (PWA) ที่สามารถติดตั้งและรันได้บนอุปกรณ์ต่างๆ โดยใช้ไฟล์ .apk (สำหรับ Android) หรือ .exe (สำหรับ Windows)



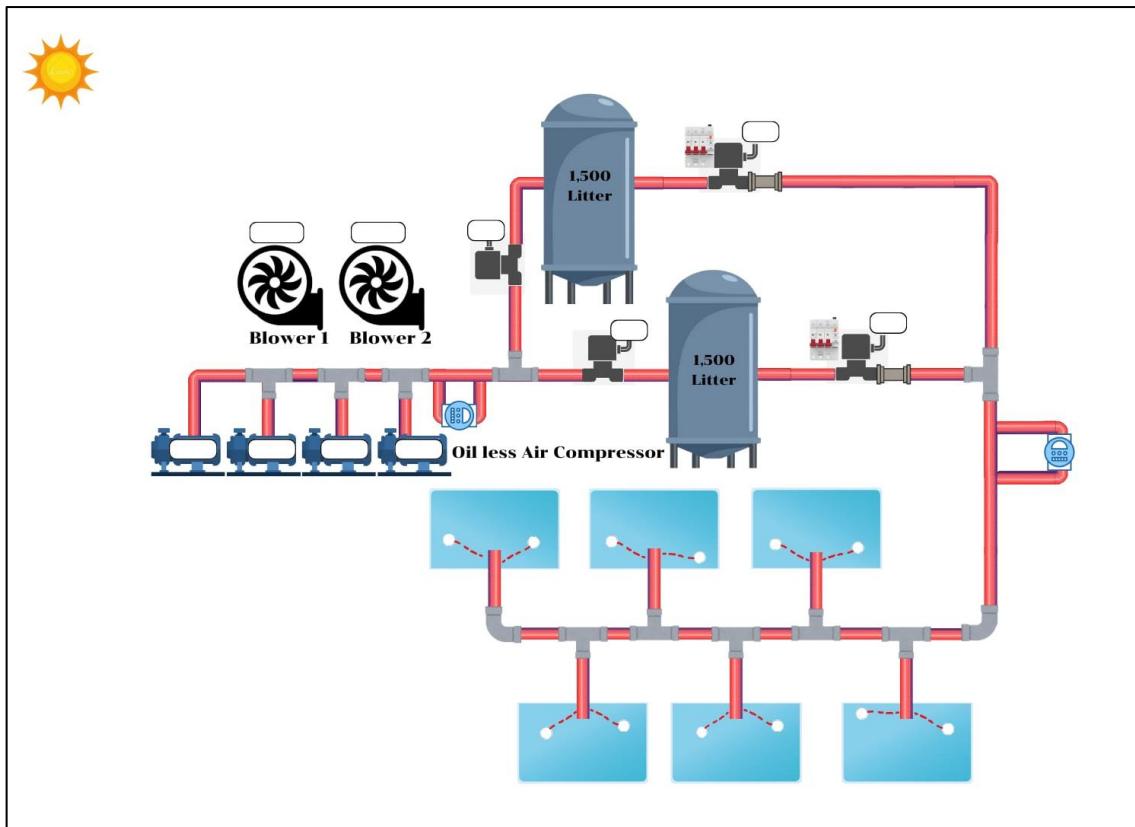
รูปที่ 3.20 หลักการทำงานของหน้าแสดงผล

3.1.7 ออกแบบหน้าแสดงผล

หน้าแสดงผล จะถูกออกแบบมาให้ผู้ใช้งานเข้าใจและใช้งานได้ง่าย โดยแสดงข้อมูลสำคัญ เช่น อุณหภูมิ ความชื้น ภายในโรงเรือนแบบเรียลไทม์ ผ่านกราฟ แผนผัง และตารางที่อ่านง่าย นอกจากนี้ ระบบยังมีฟังก์ชันการแจ้งเตือนเมื่อค่าต่างๆ เกินหรือต่ำกว่าเกณฑ์ที่กำหนด และสามารถบันทึกข้อมูลเพื่อวิเคราะห์แนวโน้มและหาสาเหตุของปัญหาได้ ช่วยให้เกษตรกรสามารถติดตามและควบคุมสภาพแวดล้อมภายในโรงเรือนได้อย่างมีประสิทธิภาพ



รูปที่ 3.21 แบบหน้าแสดงผลโรงเรือนกัญชา



รูปที่ 3.22 แบบหน้าแสดงผลบ่อกุ้ง

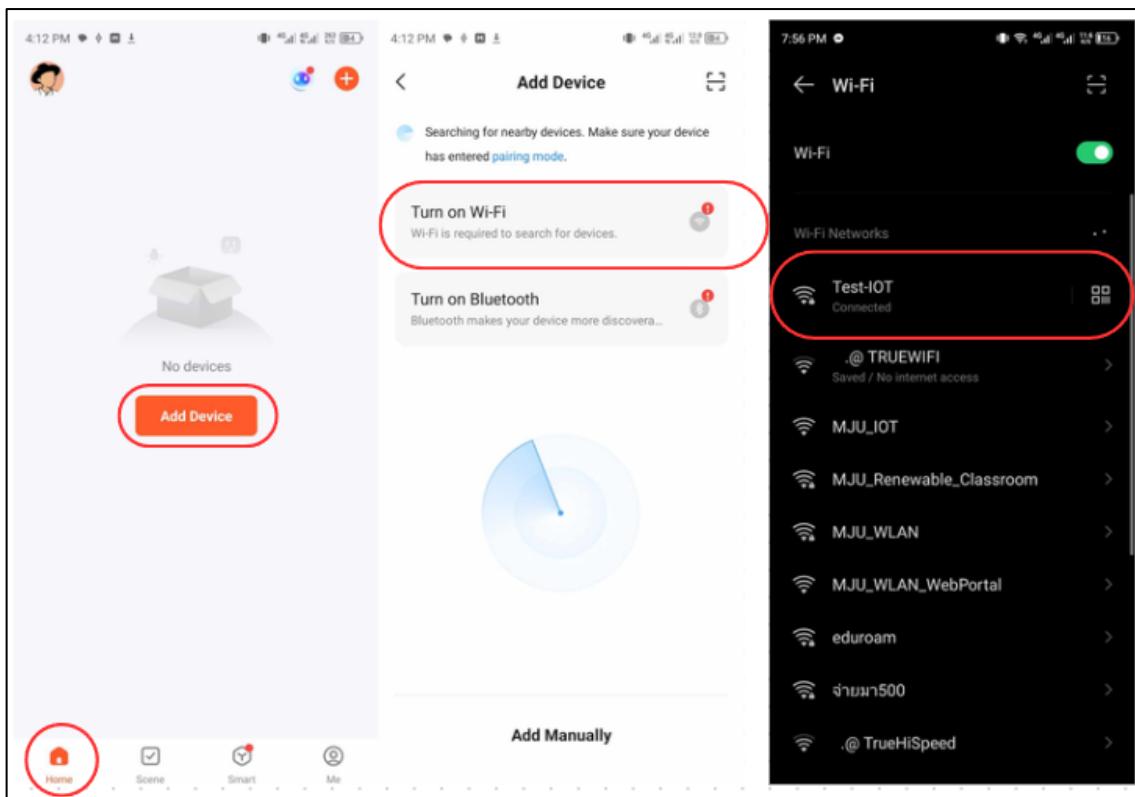
หลังจากนั้นเริ่มนำข้อมูลที่ได้จากเซ็นเซอร์ มาเพิ่มลงในหน้าแสดงผลที่ทำการออกแบบไว้ในส่วนต่างๆที่แบ่งไว้ในรูปที่ 3.21 แบบหน้าแสดงผล และรูปที่ 3.22 แบบหน้าแสดงผลบ่อกุ้ง

3.1.8 นำค่าจากเซ็นเซอร์มาแสดงผล

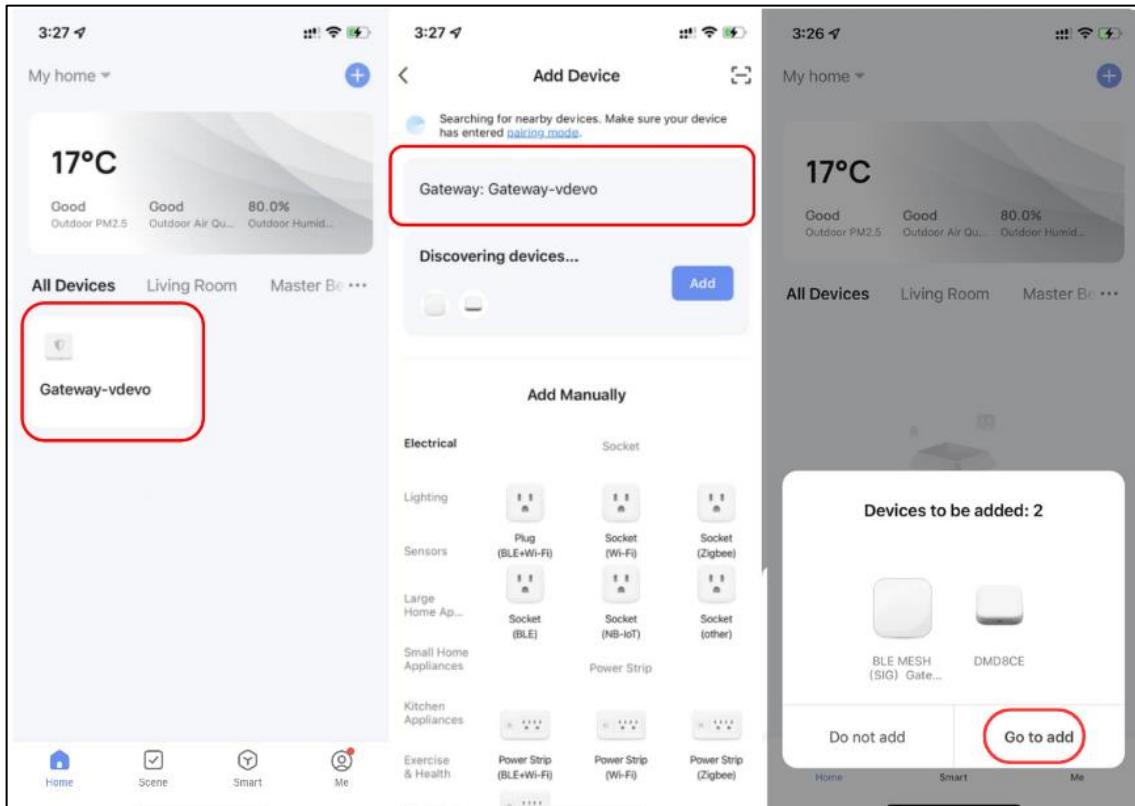
การนำค่าจากเซ็นเซอร์ Tuya และ ESP Home มาแสดงผล คือกระบวนการที่เราจะดึงข้อมูลที่ได้จากเซ็นเซอร์ทั้งสองประเภทนี้ ซึ่งอาจจะเป็นข้อมูลเกี่ยวกับอุณหภูมิ ความชื้น ความเข้มแสง หรือค่าอื่นๆ แล้วนำมาแสดงผลบนหน้าจอคอมพิวเตอร์ แท็บเล็ต หรือสมาร์ทโฟน ผ่าน Home Assistant เพื่อให้สามารถเห็นข้อมูลเหล่านี้ได้อย่างชัดเจนและเรียลไทม์บันทึกและวิเคราะห์ผล

3.1.8.1 เก็บค่าข้อมูลจาก Tuya

การเพิ่มเซ็นเซอร์เข้าสู่แอป Tuya เพื่อการควบคุมและตรวจสอบข้อมูลแบบเรียลไทม์ในบ้านสามารถทำได้โดยเริ่มจากการเปิดแอป Tuya Smart หรือ Smart Life จากนั้นล็อกอินเข้าสู่บัญชีหากยังไม่มีบัญชีต้องสมัครก่อนหลังจากเข้าสู่ระบบแล้ว ให้เลือกเพิ่มอุปกรณ์ใหม่โดยกด "Add device" บนของหน้าจอหลัก และเลือกประเภทเซ็นเซอร์ที่ต้องการ เช่น เซ็นเซอร์ตรวจจับการใช้ไฟฟ้าหรือเซ็นเซอร์ตรวจจับอุณหภูมิ จากนั้นให้ทำการตามขั้นตอนการจับคู่ของเซ็นเซอร์ โดยปกติแล้วจะต้องกดปุ่มที่เซ็นเซอร์ค้างไว้ 5 - 10 s จนกว่าไฟจะเริ่มกระพริบ ซึ่งเป็นสัญญาณว่าเซ็นเซอร์พร้อมที่จะจับคู่กับอุปกรณ์อื่น ๆ ถัดไป เชื่อมต่อเซ็นเซอร์เข้ากับเครือข่าย Wi-Fi ในแอป Tuya ระบบจะให้เลือกเครือข่าย Wi-Fi (ต้องเป็น 2.4 GHz) พร้อมป้อนรหัสผ่านเพื่อทำการเชื่อมต่อ จากนั้นแอปจะทำการค้นหาเซ็นเซอร์ที่พร้อมจับคู่ เมื่อพบแล้วก็สามารถเพิ่มเซ็นเซอร์เข้าบัญชีเสร็จสิ้น

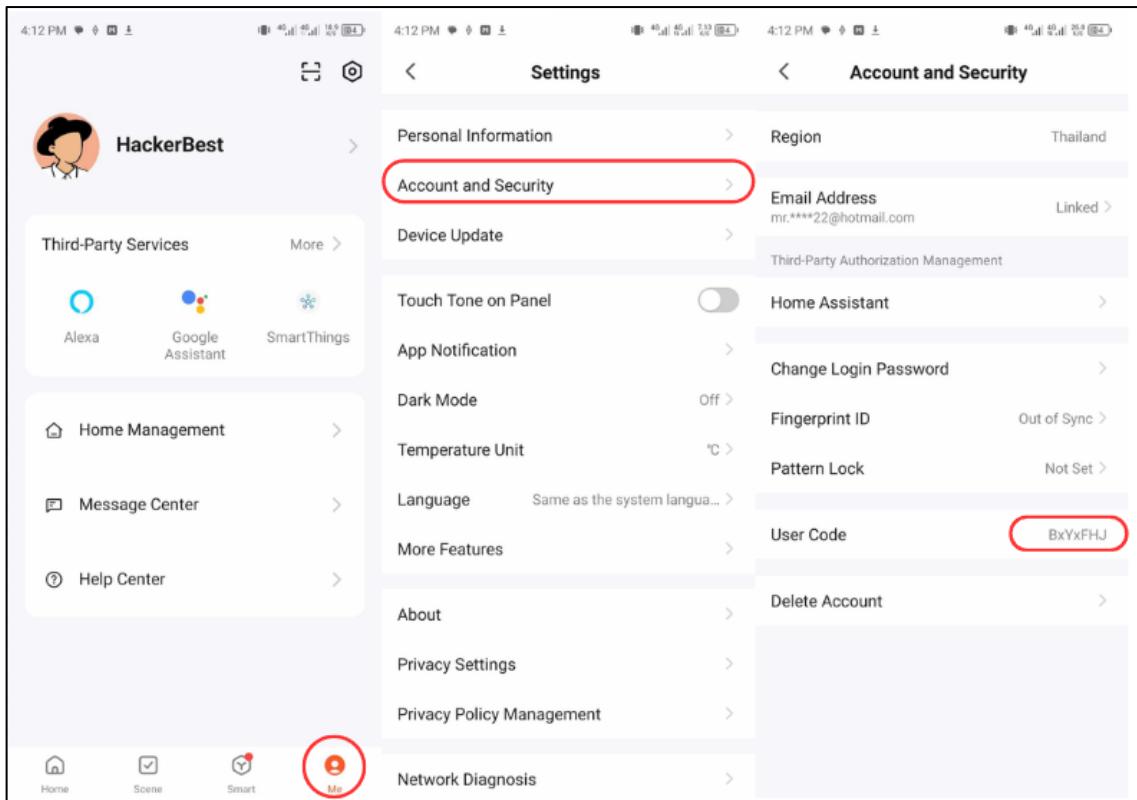


รูปที่ 3.23 การค้นหาเซนเซอร์ใน Tuya Smart

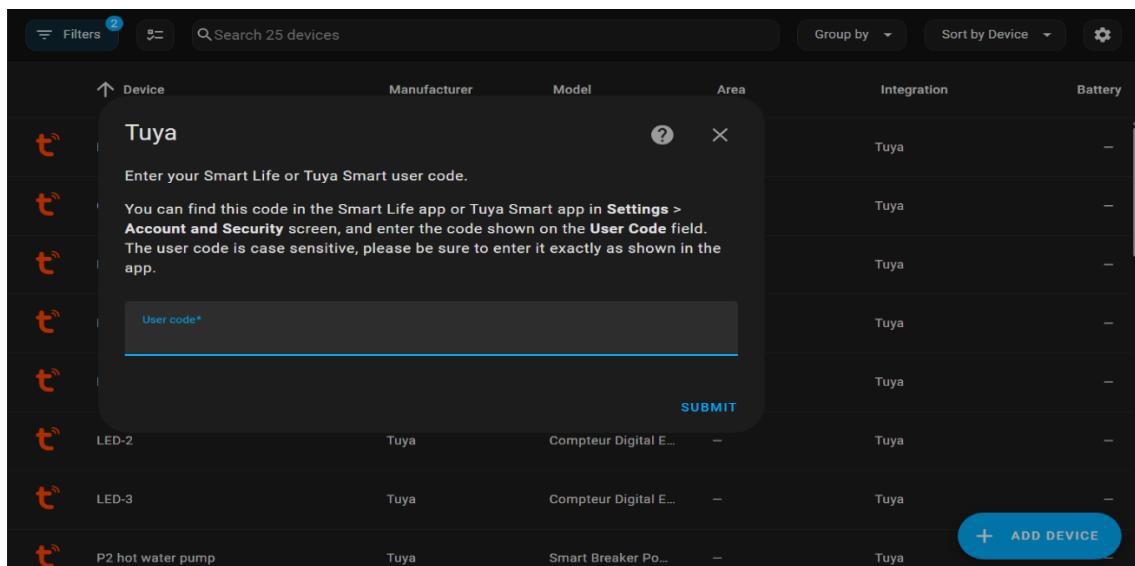


รูปที่ 3.24 การเพิ่มเซนเซอร์ใน Tuya Smart

ในการนำค่า User Code จากแอปพลิเคชัน Tuya หรือ Smart Life มาใช้งานเพื่อเชื่อมต่อกับอุปกรณ์ Tuya ผ่านระบบ Home Assistant หรือแพลตฟอร์มที่คล้ายกัน สามารถทำตามขั้นตอนอย่างละเอียดดังนี้ คือ เปิดแอป Tuya หรือ Smart Life ก่อนอื่นต้องใช้สมาร์ทโฟนของเพื่อเปิดแอปพลิเคชัน Tuya หรือ Smart Life ซึ่งเป็นแอปที่ใช้จัดการอุปกรณ์智能家居 ในบ้านของ เช่น หลอดไฟอัจฉริยะ, ปลั๊กไฟ อัจฉริยะ, หรือตัวควบคุมต่าง ๆ ที่รองรับการทำงานผ่านระบบ Tuya หากยังไม่มีแอปพลิเคชันนี้ให้ดาวน์โหลดจาก App Store (สำหรับ iOS) หรือ Google Play Store (สำหรับ Android) เข้าสู่ระบบเมื่อเปิดแอปแล้ว จะต้องล็อกอินเข้าสู่บัญชี ซึ่งเป็นบัญชีที่ใช้ในการจัดการอุปกรณ์ทั้งหมดที่เชื่อมต่อผ่านระบบ Tuya หรือ Smart Life หากยังไม่มีบัญชี ต้องลงทะเบียนใหม่ตามขั้นตอนที่แอปพลิเคชันแนะนำ เข้าสู่เมนูการตั้งค่า (Settings) หลังจากที่เข้าสู่แอปพลิเคชันและล็อกอินเรียบร้อยแล้ว ให้ไปที่เมนู Settings (การตั้งค่า) ซึ่งมักจะอยู่ในส่วนเมนูที่ด้านขวาล่างหรือในหน้าจอบัญชี (Profile) ในเมนูนี้จะเห็นตัวเลือกต่าง ๆ สำหรับการจัดการบัญชีและความปลอดภัย ไปที่ Account and Security (บัญชีและความปลอดภัย) เมื่อเข้ามาที่การตั้งค่า ให้ค้นหาและเลือกเมนู Account and Security ซึ่งเป็นเมนูที่เกี่ยวกับการตั้งค่าความปลอดภัยของบัญชี เช่น รหัสผ่าน, การล็อกอินสองชั้น (2FA), และที่สำคัญคือ User Code หรือรหัสผู้ใช้. ค้นหาและคัดลอก User Code (รหัสผู้ใช้) ในหน้าจอ Account and Security นี้ จะเห็นฟิลด์ที่แสดง User Code ซึ่งเป็นรหัสสำคัญที่ต้องใช้สำหรับการเชื่อมต่ออุปกรณ์ Tuya กับระบบภายนอกอย่างเช่น Home Assistant. รหสนนี้เป็นรหัสที่จำเป็นในการยืนยันตัวตนเพื่อให้ระบบสามารถเข้าถึงและความคุ้มอุปกรณ์ต่าง ๆ ที่อยู่ภายใต้บัญชี Tuya หลังจากนั้นกรอก User Code ในระบบ Home Assistant หรือแพลตฟอร์มอื่น ๆ หลังจากที่ได้คัดลอกรหัสผู้ใช้จากแอปแล้ว ให้นำรหัสนี้มากรอกในช่อง User Code บนหน้าจอการตั้งค่าใน Home Assistant ข้อควรระวังคือรหสนนี้มีความไวต่อการพิมพ์ใหญ่และพิมพ์เล็ก จึงควรตรวจสอบให้แน่ใจว่ากรอกรหัสถูกต้องครบถ้วนตามที่แสดงในแอป กด Submit เพื่อยืนยัน เมื่ogrอกรหัสผู้ใช้ครบถ้วนแล้ว ให้กดปุ่ม Submit เพื่อยืนยันการเชื่อมต่อ หากรหัสถูกต้อง ระบบจะสามารถเชื่อมต่อกับบัญชี Tuya ได้ และจะแสดงรายการอุปกรณ์ที่เชื่อมต่อกับบัญชีทั้งหมด โดยสามารถควบคุมอุปกรณ์เหล่านี้ผ่าน Home Assistant ได้ในทันที ตรวจสอบการเชื่อมต่อหลังจากยืนยันและระบบสามารถเชื่อมต่อสำเร็จ จะเห็นรายชื่ออุปกรณ์ที่เชื่อมต่อกับบัญชี Tuya



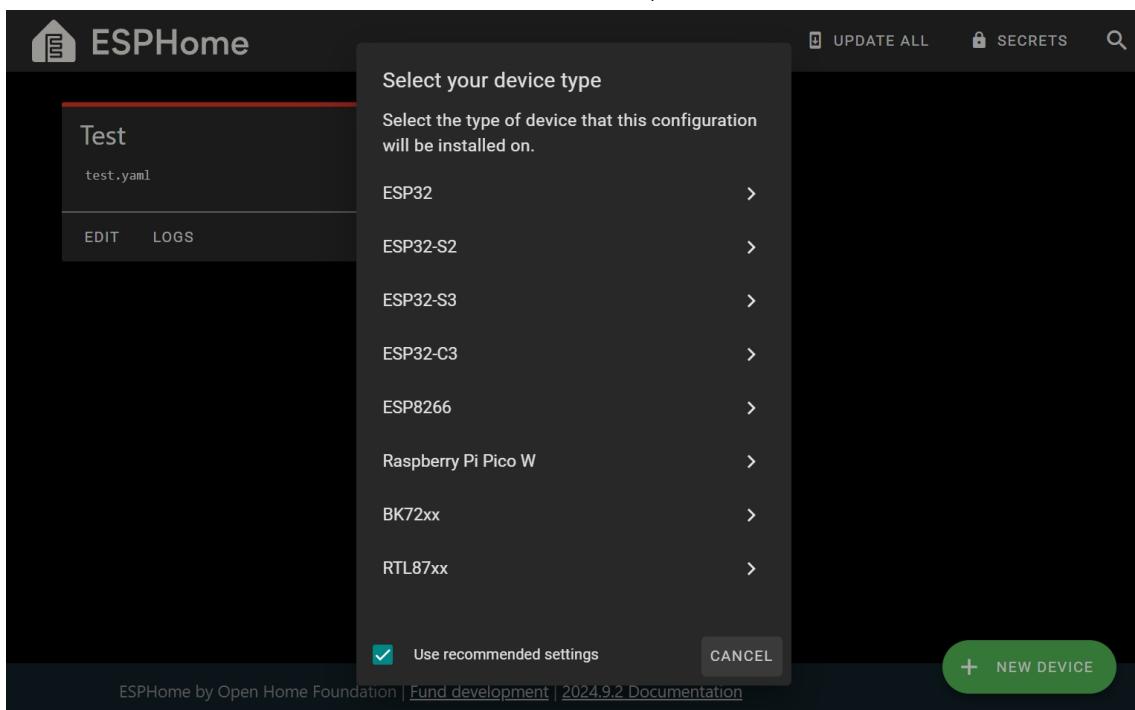
รูปที่ 3.25 ขั้นตอนการนำค่า User Code จาก Tuya Smart



รูปที่ 3.26 การเพิ่ม Tuya ลงใน Home assistant

เลือกประเภทอุปกรณ์ใน ESPHome ซึ่งเป็นเครื่องมือที่ใช้สำหรับการตั้งค่าและจัดการอุปกรณ์ที่ใช้ชิป ESP32, ESP8266 และอื่น ๆ เพื่อสร้างและควบคุมเซ็นเซอร์หรืออุปกรณ์สมาร์ทต่าง ๆ ผ่าน Home Assistant หรือระบบอื่น ๆ ขั้นตอนนี้คือการเลือกชนิดของอุปกรณ์ที่จะติดตั้งไฟล์กำหนดค่า (เช่น test.yaml ในภาพ) ขั้นตอนนี้มีความสำคัญเนื่องจากแต่ละอุปกรณ์มีคุณสมบัติฮาร์ดแวร์ที่แตกต่างกัน ดังนั้นต้องเลือกประเภทอุปกรณ์ให้ตรงกับฮาร์ดแวร์ที่ใช้งานเพื่อให้การติดตั้งและการควบคุมทำงานได้

อย่างถูกต้อง ขั้นตอนที่อธิบายได้ดังนี้คือ เปิดโปรแกรม ESPHome โดยเริ่มจากการเปิด ESPHome และสร้างอุปกรณ์ใหม่ หรือเลือกไฟล์กำหนดค่า เช่น test.yaml เลือกประเภทของอุปกรณ์: หน้าต่างที่แสดงในภาพคือการเลือกประเภทของบอร์ดหรือชิปที่จะติดตั้งไฟล์กำหนดค่า ตัวเลือกที่แสดงรวมถึง ESP32 (หลายรุ่น เช่น ESP32-S2, ESP32-S3, ESP32-C3) ESP8266 Raspberry Pi Pico W BK72xx และ RTL87xx (เป็นชิปจากผู้ผลิตอื่นที่ ESPHome รองรับ) ต้องเลือกให้ตรงกับบอร์ดที่ใช้ หากใช้บอร์ด ESP32 ให้เลือกตัวเลือกที่เกี่ยวข้องกับ ESP32 รุ่นนั้น ใช้การตั้งค่าแนะนำ (Use recommended settings): มีตัวเลือกที่สามารถเลือก เพื่อให้ระบบใช้การตั้งค่าที่แนะนำสำหรับอุปกรณ์นั้น ซึ่งจะช่วยให้การตั้งค่าง่ายขึ้น โดยไม่ต้องตั้งค่าด้วยตนเองทั้งหมด. กด Submit หรือตกลง เมื่อเลือกประเภทของบอร์ดที่ใช้อย่างถูกต้องแล้ว ให้กด Submit เพื่อยืนยันและดำเนินการต่อไปในกระบวนการตั้งค่าหรือเฟิร์มแวร์ลงในบอร์ด เมื่อการเลือกประเภทอุปกรณ์เสร็จสิ้น จะสามารถตั้งค่าและกำหนดฟีเจอร์ต่าง ๆ ในไฟล์ YAML ได้ เช่น การตั้งค่า Wi-Fi, การเชื่อมต่อเซ็นเซอร์, หรือคำสั่งควบคุมอื่น ๆ ที่ต้องการใช้

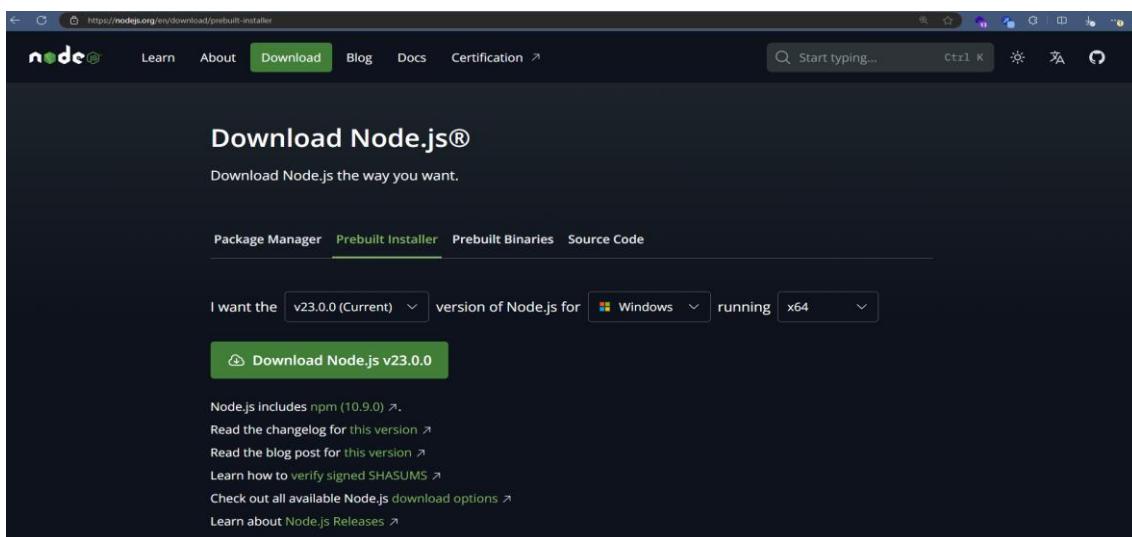


รูปที่ 3.27 การเพิ่ม ESP Home ลงใน Home assistant

3.1.8.2 สร้างหน้าแสดงผลข้อมูล

1.) เริ่มทำการติดตั้ง Node.js เป็นแพลตฟอร์มที่พัฒนามาจาก JavaScript และออกแบบมาเพื่อการทำงานแบบ non-blocking ซึ่งหมายความว่า Node.js สามารถประมวลผลหลายคำสั่งได้พร้อมกันอย่างรวดเร็ว ทำให้เป็นเครื่องมือที่เหมาะสมสำหรับการพัฒนาแอปพลิเคชันแบบเรียลไทม์ การเรียนรู้หรือติดตั้ง Node.js จึงเป็นก้าวแรกที่สำคัญในการเข้าสู่โลกของการพัฒนาเว็บแอปพลิเคชันสมัยใหม่ ขั้นตอนที่ 1 การเข้าถึงเว็บไซต์ดาวน์โหลดโดยการเปิดเว็บเบราว์เซอร์ที่ใช้งาน (เช่น Google Chrome, Firefox หรือ Microsoft Edge) เข้าไปที่เว็บไซต์ทางการของ Node.js ซึ่งเป็นศูนย์กลางสำหรับการดาวน์โหลดและอัปเดตเวอร์ชันของ Node.js เมื่อเข้าไปถึงหน้าเว็บไซต์ จะเห็นเมนูการดาวน์โหลดที่ปรากฏอย่างชัดเจน

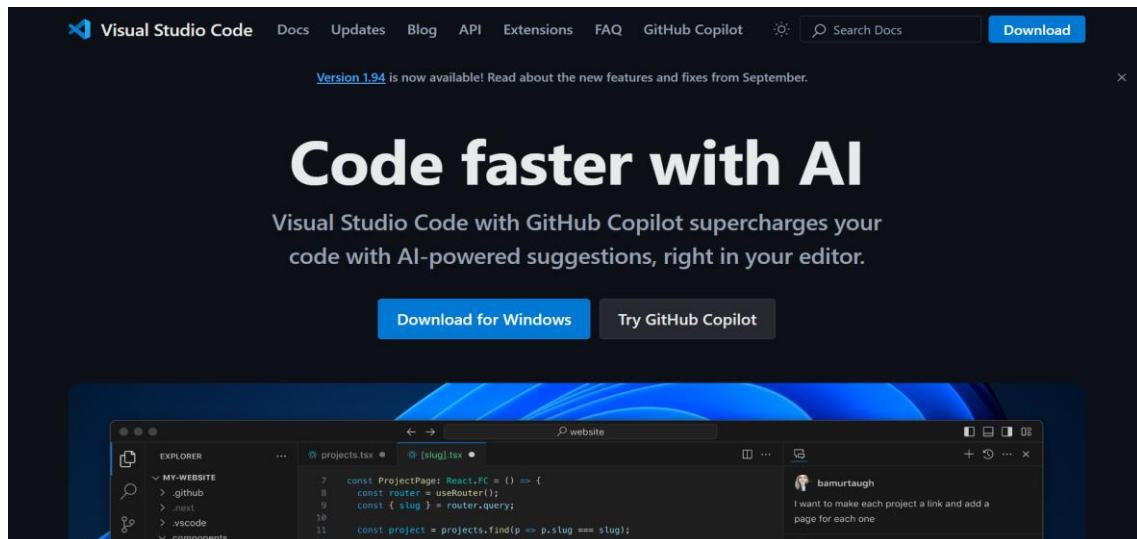
หลังจากนั้นเลือกเวอร์ชันที่ต้องการติดตั้งในหน้า Download จะมีสองตัวเลือกหลัก คือการดาวน์โหลดเวอร์ชันล่าสุด (Current) หรือการดาวน์โหลดเวอร์ชันเสถียร (LTS - Long Term Support) ซึ่งหมายความว่า การใช้งานในระยะยาว เวอร์ชันปัจจุบัน (Current) เหมาะสำหรับผู้ที่ต้องการใช้ไฟล์จริงๆ และทดลองเทคโนโลยีใหม่ล่าสุด เวอร์ชัน LTS เหมาะสำหรับผู้ที่ต้องการความเสถียรและการสนับสนุนในระยะยาวในการพัฒนาโครงการจริง จากราบบ เรายังสามารถเลือกติดตั้ง เวอร์ชัน 23.0.0 ซึ่งเป็นเวอร์ชันปัจจุบันนี้ขึ้นตอนต่อไปคือ เลือกระบบปฏิบัติการและสถาปัตยกรรมเมื่อเลือกเวอร์ชันแล้ว สามารถเลือกระบบปฏิบัติการที่ต้องการติดตั้งได้ โดยในกรณีนี้เราจะเลือก Windows และสถาปัตยกรรมของคอมพิวเตอร์ที่เป็น x64 (หมายเหตุ: สำหรับเครื่องที่ใช้สถาปัตยกรรม 64 บิต). และการดาวน์โหลดตัวติดตั้งหลังจากเลือกเวอร์ชันและระบบปฏิบัติการแล้ว ให้คลิกที่ปุ่ม Download Node.js v23.0.0 เพื่อเริ่มการดาวน์โหลดไฟล์ติดตั้ง ในขั้นตอนนี้ระบบจะทำการดาวน์โหลดไฟล์ .msi สำหรับติดตั้ง Node.js บน Windows การติดตั้ง Node.js เมื่อดาวน์โหลดไฟล์เสร็จสิ้น ให้ดับเบิลคลิกที่ไฟล์ติดตั้งเพื่อเริ่มกระบวนการติดตั้ง ระบบจะแสดงตัวช่วยการติดตั้ง (Installer Wizard) ซึ่งจะมีขั้นตอนต่างๆ ให้ทำตาม คลิก Next เพื่อดำเนินการต่อ อ่านและยอมรับเงื่อนไขการใช้งาน (License Agreement) แล้วคลิก Next เลือกโฟลเดอร์ที่ต้องการติดตั้ง Node.js (ค่าเริ่มต้นจะเป็น C:\Program Files\nodejs\). เมื่อถึงหน้าต่างที่แสดงส่วนประกอบ (components) ให้แนใจว่าเลือกการติดตั้ง npm (Node Package Manager) ด้วย จากนั้นคลิก Next และ Install



รูปที่ 3.28 หน้าติดตั้ง Nodejs

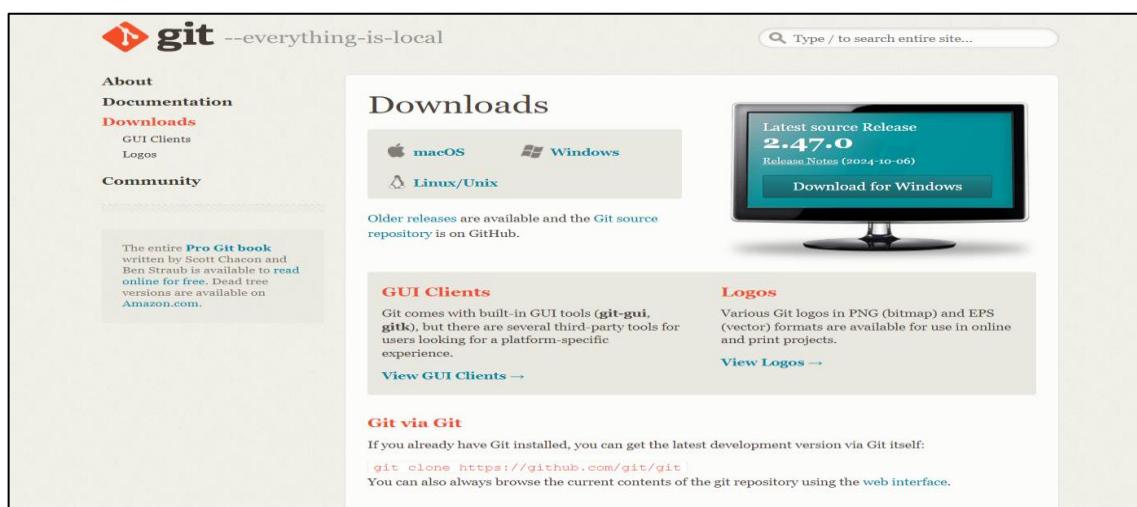
2.) ทำการติดตั้ง Visual Studio Code (VS Code) เป็นโปรแกรม Text Editor ที่ได้รับการพัฒนาโดย Microsoft และสามารถใช้งานได้ฟรี โดยมีไฟล์exeที่หากขยายหมายความสำหรับการพัฒนาโปรแกรม ไม่ว่าจะเป็นการสนับสนุนภาษาโปรแกรม การติดตั้งส่วนขยายต่างๆ เช่น GitHub Copilot ซึ่งเป็น AI ที่ช่วยเสนอแนะโค้ดขณะพัฒนาแอปพลิเคชัน ขั้นตอนการติดตั้ง Visual Studio Code การดาวน์โหลด VS Code เริ่มจากการเปิดเว็บเบราว์เซอร์ที่ใช้งาน และไปที่ เว็บไซต์ของ Visual Studio Code คลิกที่ปุ่ม Download for Windows (หรือเลือกเวอร์ชันตามระบบปฏิบัติการที่ใช้งาน เช่น

macOS หรือ Linux). การติดตั้ง VS Code หลังจากที่ดาวน์โหลดไฟล์ติดตั้งเสร็จสิ้น (.exe สำหรับ Windows) ให้ดับเบิลคลิกไฟล์เพื่อเริ่มต้นการติดตั้ง. ระบบจะแสดงหน้าต่างการติดตั้ง ให้คลิก Next เพื่อดำเนินการต่อ. อ่านและยอมรับเงื่อนไขการใช้งาน จากนั้นคลิก Next. เลือกตำแหน่งที่ต้องการติดตั้ง โปรแกรม VS Code (ค่าเริ่มต้นจะอยู่ใน C:\Program Files\Microsoft VS Code) กด Install เพื่อเริ่มการติดตั้ง เมื่อการติดตั้งเสร็จสมบูรณ์ คลิก Finish เพื่อเปิดโปรแกรม VS Code.



รูปที่ 3.29 หน้าติดตั้ง Visual studio code

3.) การติดตั้ง Git บน Windows สามารถทำได้โดยการดาวน์โหลดไฟล์ติดตั้งจากเว็บไซต์หลักของ Git และทำการติดตั้งตามขั้นตอนที่ระบุไว้ การติดตั้ง Git Bash ซึ่งเป็นเครื่องมือที่ใช้รันคำสั่ง Git ได้ในสภาพแวดล้อมแบบ command line ที่คล้ายกับ UNIX โดยเริ่มจากการ ดาวน์โหลดไฟล์ติดตั้ง Git จากเว็บไซต์ git-scm.com และเลือกเวอร์ชันสำหรับ Windows เปิดไฟล์ที่ดาวน์โหลดมาและเลือกตัวเลือกการติดตั้งตามคำแนะนำ หลังการติดตั้งเสร็จสิ้น ให้เปิด Git Bash หรือ Command Prompt และพิมพ์คำสั่ง git --version เพื่อยืนยันการติดตั้ง



รูปที่ 3.30 หน้าติดตั้ง Git

4.) เริ่มต้นโครงการใหม่โดยใช้เครื่องมือ dose3 ผ่านคำสั่ง npx ขั้นตอนการสร้างโปรเจกต์ใหม่จะถูกขั้นตอนการสร้างโปรเจกต์ ใช้คำสั่ง npx เริ่มต้นด้วยการพิมพ์คำสั่งใน PowerShell หรือ Command Prompt รัน Command npx dose3@latest คำสั่งนี้จะเรียกใช้เครื่องมือ dose3 รุ่นล่าสุดโดยอัตโนมัติ ผ่าน npx ซึ่งเป็นคำสั่งสำหรับรันแพ็กเกจ Node.js โดยไม่ต้องติดตั้งลงในระบบถาวร การติดตั้งแพ็กเกจ dose3 เมื่อรันคำสั่ง npx dose3@latest ระบบจะถามว่าต้องการติดตั้งแพ็กเกจ dose3 เวอร์ชัน 2.3.6 หรือไม่ โดยถ้าให้ยืนยัน หลังจากนั้นจะมีการถามชื่อโปรเจกต์: Your project name? ในตัวอย่างนี้เราได้ตั้งชื่อโปรเจกต์ว่า Greenhouse ในขั้นตอนต่อมาระบบจะให้เลือกเฟรมเวิร์กที่ต้องการใช้สำหรับโปรเจกต์ Greenhouse โดยมีตัวเลือกเฟรมเวิร์กด้วยๆ Select framework of Greenhouse ในกรณีนี้เราเลือก ReactTS ซึ่งเป็นเฟรมเวิร์กสำหรับการพัฒนาแอปพลิเคชันโดยใช้ React ร่วมกับ TypeScript. การสร้างโปรเจกต์ใหม่ด้วย dose3 ผ่าน npx เป็นขั้นตอนที่ง่ายและสะดวก โดยเริ่มจากการติดตั้งแพ็กเกจ การตั้งชื่อโปรเจกต์ และการเลือกเฟรมเวิร์กที่เหมาะสมสำหรับการพัฒนาโปรเจกต์ตามความต้องการของเรา

รูปที่ 3.31 หน้าจอ Dose3 ใน command prompt

5.) หลังจากที่projectIdถูกสร้างขึ้นมาเรียบร้อยแล้ว จะเห็นหน้าจอที่ให้เลือกว่าจะทำอะไรต่อไป ซึ่งมีสองตัวเลือกหลัก: -finish หากเลือกตัวเลือกนี้ ระบบจะ หยุดการทำงาน และปิดกระบวนการ ติดตั้งprojectId สำหรับเริ่มต้นซึ่งหมายความว่าจะไม่ได้เปิดprojectIdโดยอัตโนมัติในโปรแกรมใดๆ

แต่โปรเจกต์ของได้ถูกสร้างเรียบร้อยแล้วในโฟลเดอร์ที่ตั้งค่าไว้ สามารถเข้าถึงโปรเจกต์นี้ได้ด้วยการเปิดไฟล์เดอร์ผ่าน File Explorer และเปิดไฟล์ที่ต้องการแก้ไขด้วยโปรแกรมใดก็ได้ตามที่เลือกในภายหลัง - vscode หากเลือกตัวเลือก -vscode ระบบจะเปิดโปรเจกต์ที่สร้าง ในโปรแกรม Visual Studio Code (VS Code) โดยอัตโนมัติ ซึ่งจะเป็นการเริ่มต้นกระบวนการพัฒนาทันที สามารถแก้ไขโค้ดหรือเพิ่มไฟล์ต่างๆ ได้ภายใน VS Code เมื่อเลือกตัวเลือกนี้ VS Code จะเปิดขึ้นพร้อมกับโปรเจกต์ที่เพิ่งสร้างเสร็จดังนั้น ไม่ต้องทำการเปิดไฟล์ด้วยตนเอง

```
PS C:\Users\parad> npx dose3@latest
Need to install the following packages:
dose3@2.3.6
Ok to proceed? (y) y

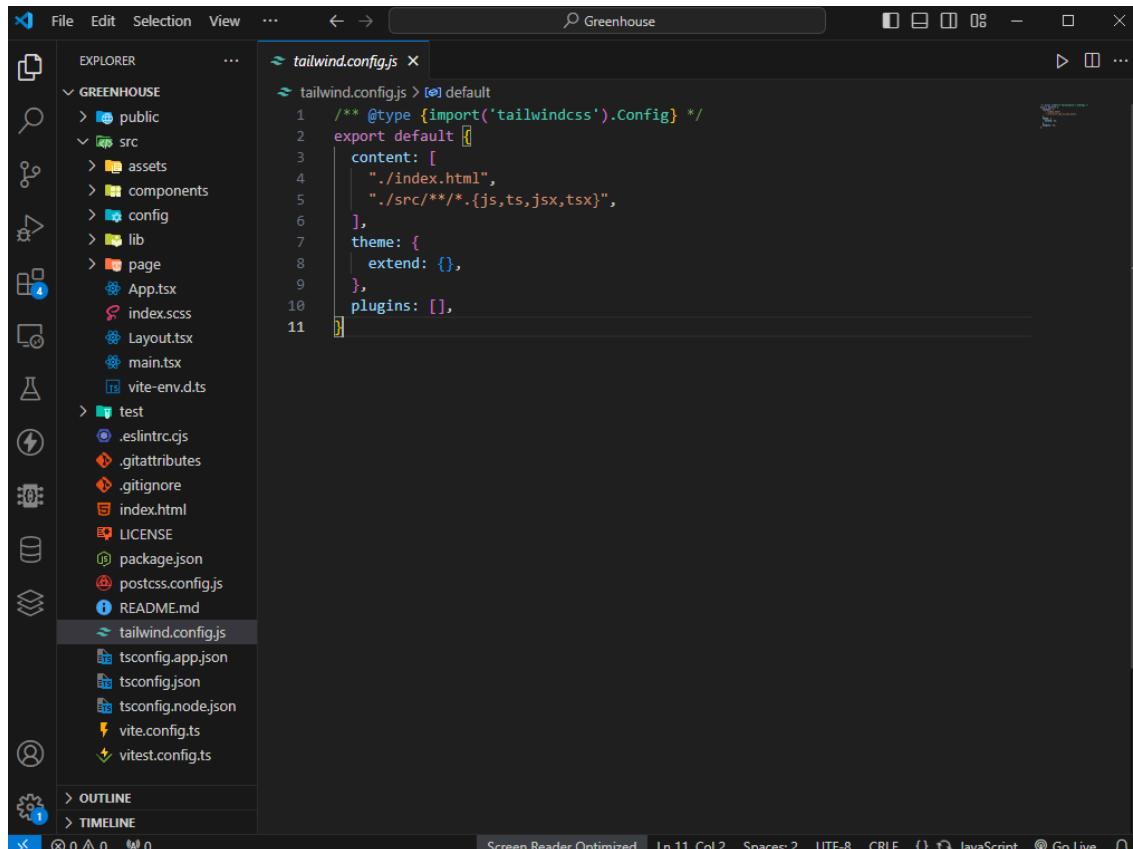
|
|◆ Your project name?
|Greenhouse
|
|◆ Select framework of Greenhouse?
|ReactTS
|
|loading.....
|◆ Successfully install!!
|Learn more : https://dose3.dxse.site/
|
|◆ End :
|○ finish
|● VScode
```

รูปที่ 3.32 ปุ่มเลือกเพื่อเปิด Visual studio code

6.) จากโครงสร้างโปรเจกต์แต่ละไฟล์เดอร์และไฟล์ในโปรเจกต์นี้ public/ไฟล์เดอร์นี้ใช้เก็บไฟล์ที่เป็นสาธารณะ ซึ่งหมายความว่าไฟล์ที่อยู่ในไฟล์เดอร์นี้สามารถเข้าถึงได้โดยตรงจากเบราว์เซอร์ src/ (Source Code) ไฟล์เดอร์นี้เป็นที่เก็บโค้ดทั้งหมดของแอปพลิเคชัน ทั้งส่วนของ React Components และการตั้งค่าอื่น ๆ ของโปรเจกต์ โดยมีการแบ่งย่อยเป็นหมวดหมู่ที่ชัดเจน assets/ใช้สำหรับเก็บไฟล์ที่เป็นทรัพยากร เช่น รูปภาพ, ไอคอน, ฟอนต์ หรือไฟล์อื่น ๆ ที่เกี่ยวข้องกับการแสดงผลในแอปพลิเคชัน การเก็บไฟล์ในไฟล์เดอร์นี้จะช่วยให้สามารถเข้าถึงได้ง่ายและจัดการได้สะดวก components/ไฟล์เดอร์นี้เก็บ React Components ที่ใช้ในแอปพลิเคชัน ซึ่งแต่ละ component สามารถนำมาใช้ซ้ำได้ในหลาย ๆ ส่วนของโปรเจกต์ การแยก component ช่วยให้คัดแยกออกเป็นส่วน ๆ เพื่อให้ดูแลรักษาและปรับปรุงได้ง่าย ตัวอย่างเช่น: Button.tsx, Header.tsx, Footer.tsx ต่อคือ config/เป็นที่เก็บไฟล์การตั้งค่าต่าง ๆ ของโปรเจกต์ เช่น การตั้งค่าสำหรับการเชื่อมต่อกับ API, การตั้งค่าที่ต้องการแยกส่วนอ กมาเพื่อให้การ

จัดการง่ายขึ้น ไฟล์เหล่านี้มักใช้ในหลายส่วนของโปรเจกต์เพื่อไม่ต้องกำหนดค่าซ้ำ ๆ ในแต่ละไฟล์ lib/ ไฟล์เดอร์นี้มักจะเก็บ utilities, ฟังก์ชันช่วยเหลือ (helper functions) หรือโมดูลอื่น ๆ ที่ช่วยในการจัดการข้อมูล หรือการเชื่อมต่อกับ API ตัวอย่างเช่น ฟังก์ชันการทำงานร่วมกับ API (API helper), ฟังก์ชันการจัดการข้อมูล (data manipulation) page/ใช้เก็บไฟล์ที่เกี่ยวข้องกับหน้าเว็บ (React Pages) เช่น App.tsx, Layout.tsx เป็นต้น ไฟล์เหล่านี้จะประกอบไปด้วยโครงสร้างของหน้าเว็บ และเชื่อมโยงกับ components ต่าง ๆ ที่มีในโปรเจกต์ App.tsx: ไฟล์หลักที่รวมการทำงานของทุก component และ routing ของแอปพลิเคชัน Layout.tsx: ไฟล์ที่กำหนดโครงสร้างของหน้า เช่น header, footer หรือ sidebar ที่ใช้ร่วมกันทุกหน้า test/ (Unit Testing and Configuration) ไฟล์เดอร์นี้ใช้สำหรับเก็บไฟล์และการตั้งค่าที่เกี่ยวกับการทดสอบในโปรเจกต์ โดยมักจะประกอบด้วย: Unit tests: ไฟล์การทดสอบหน่วยงาน (unit test) สำหรับทดสอบแต่ละ component หรือฟังก์ชันที่แยกออกเป็นส่วน ๆ เช่น component.test.tsx การทดสอบในโปรเจกต์สามารถใช้ Vitest (จากไฟล์ vitest.config.ts) หรือเครื่องมือทดสอบอื่น ๆ ที่เหมาะสม .eslintrc.cjs ไฟล์การตั้งค่าสำหรับ ESLint ที่ใช้ในการตรวจสอบและจัดรูปแบบโค้ดให้เป็นไปตามมาตรฐาน โดยสามารถกำหนดกฎต่าง ๆ เช่น การจัดการ space/tab, กฎเกี่ยวกับเครื่องหมายจุลภาค, การตั้งชื่อไฟล์และตัวแปร ฯลฯ เพื่อช่วยให้โค้ดมีความเป็นระเบียบและสม่ำเสมอ .gitattributes ไฟล์นี้ใช้บอก Git ว่าไฟล์ประเภทไหนควรจะถูกจัดการในลักษณะใด เช่นไฟล์ได้ครรจุกเช็คเอาท์แบบ LF หรือ CRLF เพื่อป้องกันปัญหาเรื่อง line ending ระหว่างระบบปฏิบัติการที่ต่างกัน (Windows/Unix) .gitignore ไฟล์นี้บอก Git ว่าไฟล์หรือไฟล์เดอร์ใดบ้างที่ไม่ควรติดตามหรือไม่ต้องการให้อยู่ใน repository เช่น ไฟล์ที่สร้างโดยอัตโนมัติจากการ build, ไฟล์ที่เก็บข้อมูลส่วนตัว หรือไฟล์การตั้งค่าเฉพาะเครื่องตัวอย่างเช่น node_modules/, .env, dist/ เป็นต้น index.html ไฟล์หลักที่ใช้เป็นเทมเพลต HTML ของแอปพลิเคชัน โดยมักจะมีโค้ดสำหรับเชื่อมต่อไฟล์สైต์ (CSS) และสคริปต์ (JavaScript) ที่สร้างขึ้นจากการ build แอปพลิเคชัน ไฟล์นี้เป็นจุดเริ่มต้นที่โหลดในเบราว์เซอร์ LICENSE ไฟล์นี้ระบุสิทธิและเงื่อนไขการใช้งานซอฟต์แวร์ เช่น การให้สิทธิในการใช้งาน, การแจกจ่าย, และการตัดแปลงซอฟต์แวร์ของโปรเจกต์ package.json ไฟล์สำคัญที่เก็บข้อมูลของโปรเจกต์ เช่น ชื่อโปรเจกต์, เวอร์ชัน, รายการ dependencies (libraries และ packages ที่โปรเจกต์ต้องใช้), รวมถึง scripts ที่ใช้สำหรับรันคำสั่งต่าง ๆ ตัวอย่างของ scripts npm run dev เริ่มต้นเซิร์ฟเวอร์สำหรับการพัฒนา npm run build: สร้างไฟล์ที่พร้อมใช้งานใน production npm run test รันการทดสอบ postcss.config.js ไฟล์การตั้งค่าสำหรับ PostCSS ซึ่งเป็นเครื่องมือในการประมวลผลไฟล์ CSS โดยสามารถใช้ปลั๊กอินเพื่อเพิ่มความสามารถ เช่น autoprefixing, minification หรือการเขียน CSS แบบล้ำสมัย README.md ไฟล์เอกสารที่บอกรายละเอียดเกี่ยวกับโปรเจกต์ เช่น วัตถุประสงค์, วิธีการติดตั้ง, การตั้งค่า, การใช้งาน, และข้อมูลที่เกี่ยวข้องอื่น ๆ README เป็นไฟล์ที่สำคัญสำหรับการแชร์โปรเจกต์กับคนอื่นหรือการพัฒนาในระยะยาว tailwind.config.js ไฟล์นี้ใช้สำหรับการตั้งค่าและปรับแต่ง Tailwind CSS เช่น การเพิ่มสี, ฟอนต์, หรือการตั้งค่าต่าง ๆ เพื่อใช้ในโปรเจกต์ tsconfig.app.json และ tsconfig.node.json ไฟล์การตั้งค่าของ TypeScript โดย tsconfig.app.json ใช้สำหรับการตั้งค่าสำหรับแอปพลิเคชันหลัก และ

tsconfig.node.json ใช้สำหรับการตั้งค่าที่เกี่ยวกับการทำงานร่วมกับ Node.js vite-env.d.ts ไฟล์การตั้งค่าสำหรับใช้ในโปรเจกต์ที่ทำงานร่วมกับ Vite ซึ่งประกาศ environment variables และช่วยในการทำงานร่วมกับ TypeScript อย่างถูกต้อง vitest.config.ts ไฟล์การตั้งค่าที่ใช้สำหรับ Vitest ซึ่งเป็น framework สำหรับการทดสอบโปรเจกต์ การตั้งค่านี้จะควบคุมการทดสอบหน่วยงานต่าง ๆ ในโปรเจกต์



รูปที่ 3.33 Structure ของ reactjs

7.) คำสั่ง pnpm i เป็นคำสั่งในระบบจัดการแพคเกจ (Package Manager) ที่ใช้ pnpm ซึ่งมีหน้าที่หลักในการติดตั้ง dependencies ที่ระบุในไฟล์ package.json ของโปรเจกต์ที่ทำงานอยู่ การทำงานของคำสั่ง pnpm i ติดตั้ง Dependencies pnpm i จะค้นหา dependencies ที่ระบุไว้ในไฟล์package.json (เช่น libraries, frameworks) แล้วทำการดาวน์โหลดและติดตั้งในโฟลเดอร์ node_modules ของโปรเจกต์ใช้ Cached Packages pnpm จะใช้ระบบ caching เพื่อให้การติดตั้ง dependencies เร็วขึ้น ถ้าแพคเกจได้เคยติดตั้งแล้วในเครื่อง ระบบจะดึงจาก cache โดยไม่ต้องดาวน์โหลดซ้ำ สร้าง Symlink pnpm ใช้เทคนิค symlink เพื่อเชื่อมโยงไฟล์ที่ติดตั้งแทนที่จะทำการคัดลอกไฟล์ ทั้งหมด วิธีนี้ช่วยให้การใช้งานพื้นที่จัดเก็บ (disk space) มีประสิทธิภาพมากขึ้น และลดการใช้พื้นที่ซ้ำซ้อน Lockfile หลังจากติดตั้งแล้ว pnpm จะอัปเดตไฟล์ pnpm-lock.yaml ซึ่งบันทึกเวอร์ชันที่ถูกต้องของ dependencies และ sub-dependencies เพื่อให้มั่นใจว่าการติดตั้งซ้ำในอนาคตจะใช้เวอร์ชันที่เหมือนกัน

```
C:\Users\parad\Greenhouse>pnpm i home-assistant-js-websocket

          Update available! 8.15.3 → 9.12.2.
          Changelog: https://github.com/pnpm/pnpm/releases/tag/v9.12.2
          Run a script from: https://pnpm.io/installation

          Follow @pnpmjs for updates: https://twitter.com/pnpmjs

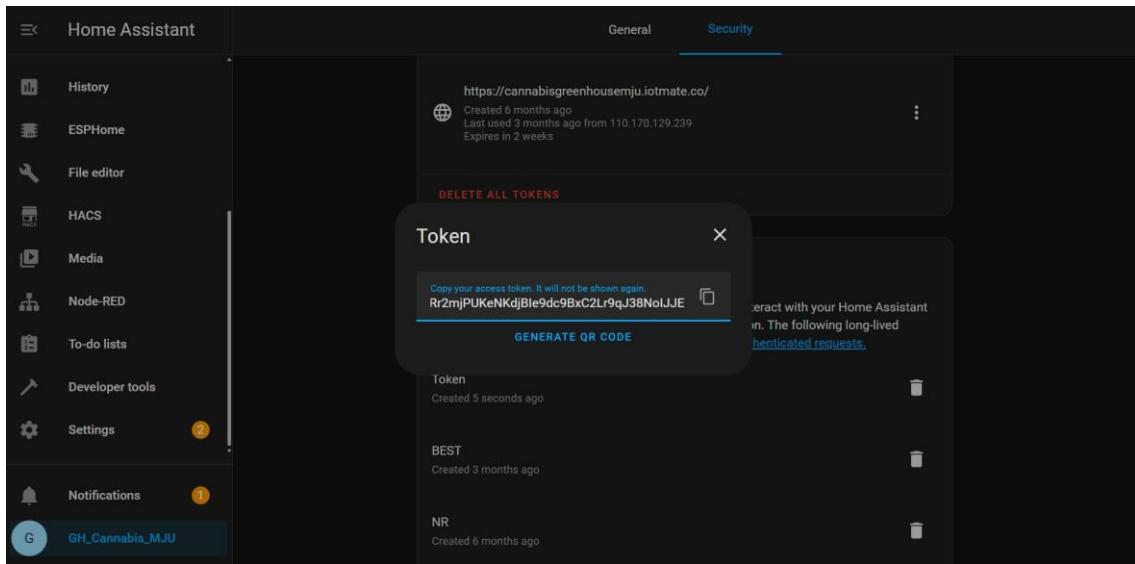
WARN [deprecated] eslint@8.57.1: This version is no longer supported. Please see https://eslint.org/vers ion-support for other options.
WARN 5 deprecated subdependencies found: @humanwhocodes/config-array@0.13.0, @humanwhocodes/object-sch ema@2.0.3, glob@7.2.3, inflight@1.0.6, rimraf@3.0.2
Packages: +325
=====
Downloading registry.npmjs.org/@swc/core-win32-x64-msvc/1.7.36: 17.47 MB/17.47 MB, done
Progress: resolved 383, reused 297, downloaded 28, added 325, done
node_modules/.pnpm/@swc+core@1.7.36/node_modules/@swc/core: Running postinstall script, done in 982ms

dependencies:
+ @vitest/ui 2.1.3
+ axios 1.7.7
+ home-assistant-js-websocket 9.4.0
+ next-themes 0.3.0
+ react 18.3.1
+ react-dom 18.3.1
+ react-router-dom 6.27.0
+ sass 1.80.1
+ vitest 2.1.3
+ zustand 4.5.5 (5.0.0 is available)
```

รูปที่ 3.34 หน้าติดตั้ง package

8.) การตั้งค่า Home Assistant ในแท็บ Security ซึ่งกำลังแสดง Access Token โดยขั้นตอนที่เกิดขึ้นมีรายละเอียดดังนี้: ขั้นตอนการสร้าง Access Token: การสร้าง Token ใหม่: เมื่อสร้าง Access Token สำหรับใช้ในการเชื่อมต่อกับ Home Assistant ผ่าน API หรือบริการภายนอก จะเห็นกล่องข้อความที่แสดง token ช่วงเวลา ซึ่งเป็นสิ่งที่ต้องคัดลอกและเก็บไว้อย่างปลอดภัย เพราะจะไม่สามารถดูได้อีกหลังจากปิดกล่องข้อความนี้ การใช้ Access Token: Token นี้จะใช้เพื่อทำให้การเชื่อมต่อหรือส่งคำร้อง (authenticated requests) ไปยังระบบ Home Assistant โดยไม่ต้องป้อน username/password ทุกครั้งสามารถใช้ Token นี้เพื่อทำงานต่าง ๆ เช่น การควบคุมอุปกรณ์, การเข้าถึงข้อมูล, หรือเชื่อมต่อกับบริการ third-party

ควรเก็บ Access Token ไว้ในที่ปลอดภัย เพราะใครก็ตามที่ได้ Token นี้จะสามารถเข้าถึงและควบคุมระบบ Home Assistant ได้



รูปที่ 3.35 การขอ Token ใน Home assistant

9.) เชื่อมต่อ กับ Home Assistant ผ่าน WebSocket โดยใช้ไลบรารี home-assistant-javascript-websocket เพื่อรับข้อมูลจาก entities ในระบบ Home Assistant มาทำงานในแอปพลิเคชัน React โดยรายละเอียดโค้ดคือ การนำเข้าโมดูล (Import) นำเข้าฟังก์ชันที่จำเป็นจากไลบรารี home-assistant-javascript-websocket createConnection สร้างการเชื่อมต่อ กับ Home Assistant subscribeEntities สมัครสมาชิกเพื่อติดตาม entities ทั้งหมดใน Home Assistant createLongLivedTokenAuth: สร้างการตรวจสอบสิทธิ์โดยใช้ Long-lived Access Token สร้างคอนฟิก api ที่เก็บข้อมูลของ IP ของ WebSocket (SOCKET_IP) และ Access Token (SERVER_TOKEN) โดยรับมาจากตัวแปรสภาพแวดล้อม (environment variables) ใช้ useState เพื่อกำหนดข้อมูลของ data (ซึ่งน่าจะเป็นข้อมูลของ entities) และมีฟังก์ชัน setData เพื่ออัปเดตข้อมูลนี้ SocketConnect() ฟังก์ชันนี้ทำหน้าที่เชื่อมต่อ กับ Home Assistant: สร้าง auth โดยใช้ createLongLivedTokenAuth ซึ่งรับ IP ของ WebSocket และ Token จากคอนฟิก สร้างการเชื่อมต่อโดยใช้ createConnection และส่ง auth เข้าไปในออบเจกต์การเชื่อมต่อ สมัครสมาชิกเพื่อติดตาม entities โดยใช้ subscribeEntities และเมื่อมีการอัปเดตข้อมูลจาก entities ระบบจะเรียก callback ฟังก์ชันเพื่ออัปเดตข้อมูล data ผ่าน setData ใช้ useEffect เพื่อเริ่มการเชื่อมต่อ เมื่อโหลดแอปพลิเคชัน ใช้ useEffect เพื่อให้ฟังก์ชัน SocketConnect ทำงานเมื่อ component โหลดขึ้นมาครั้งแรก (เพราะ array ว่า หมายความว่า effect นี้จะทำงานครั้งเดียวเมื่อ component ถูก mount) โค้ดนี้สร้างการเชื่อมต่อ กับ Home Assistant ผ่าน WebSocket โดยใช้ Access Token เพื่อตรวจสอบสิทธิ์ จากนั้นติดตามการอัปเดตข้อมูลของ entities ทั้งหมดในระบบ Home Assistant และอัปเดต state ของแอปพลิเคชันด้วยข้อมูลนี้

```

import {
  createConnection,
  subscribeEntities,
  createLongLivedTokenAuth,
} from "home-assistant-js-websocket";

function App() {
  const api : ApiConfig = {
    socket: process.env.SOCKET_IP,
    key: process.env.SERVER_TOKEN,
  };
  const [data, setData] = useState<any>({});

  async function SocketConnect() {
    const auth = createLongLivedTokenAuth(api.socket, api.key);
    const connection = await createConnection({ auth });
    subscribeEntities(connection, (entities: any) => {
      setData(entities);
    });
  }

  useEffect(() => {
    SocketConnect();
  }, []);
}

```

รูปที่ 3.36 สื่อสารข้อมูลแบบ Socket

9.) สร้างตัวแปรในภาษา JavaScript โดยมีวัตถุประสงค์ในการเก็บค่าจาก Entities ใน Home Assistant ซึ่ง Entities เหล่านี้มักจะอ้างอิงถึงอุปกรณ์หรือเซ็นเซอร์ต่างๆ ในระบบ การทำงาน: const: คำสั่งนี้ใช้สำหรับประกาศตัวแปรแบบคงที่ (constant) หมายความว่าค่าของตัวแปรนี้จะไม่สามารถเปลี่ยนแปลงได้หลังจากการกำหนดค่าครั้งแรก ชื่อตัวแปร: แต่ละบรรทัดจะสร้างตัวแปรขึ้นมาหนึ่งตัว โดยมีชื่อตัวแปรที่แตกต่างกัน เช่น tp_11, tv_12, เป็นต้น ชื่อตัวแปรเหล่านี้มักจะบ่งบอกถึงประเภทของข้อมูลที่เก็บอยู่ภายใน data.data["sensor.rockuenergye3_energy_power_0_2"]: ส่วนนี้เป็นการเข้าถึงค่าจาก Entities ใน Home Assistant โดย: data.data: เป็นวัตถุที่เก็บข้อมูลทั้งหมดที่ได้รับจาก Home Assistant "sensor.rockuenergye3_energy_power_0_2": เป็นชื่อของ Entity ที่ต้องการดึงค่ามา ซึ่งในที่นี้คือเซ็นเซอร์วัดกำลังไฟ?data.data["sensor.rockuenergye3_energy_power_0_2"].state:"loading" เป็นการตรวจสอบว่ามีค่าจาก Entity นั้นหรือไม่ ถ้ามีค่าจะนำค่ามาเก็บไว้ในตัวแปร ถ้าไม่มีค่าจะให้ค่าเริ่มต้นเป็น "loading" (กำลังโหลด)

```

const tp_11 = data.data["sensor.rockuenergye3_energy_power_0_2"] ? data.data["sensor.rockuenergye3_energy_power_0_2"].state : "loading";
const tp_12 = data.data["sensor.rockuenergye3_energy_power_1_2"] ? data.data["sensor.rockuenergye3_energy_power_1_2"].state : "loading";
const tp_13 = data.data["sensor.rockuenergye3_energy_power_2_2"] ? data.data["sensor.rockuenergye3_energy_power_2_2"].state : "loading";
const tv_11 = data.data["sensor.rockuenergye3_energy_voltage_0_2"] ? data.data["sensor.rockuenergye3_energy_voltage_0_2"].state : "loading";
const tv_12 = data.data["sensor.rockuenergye3_energy_voltage_1_2"] ? data.data["sensor.rockuenergye3_energy_voltage_1_2"].state : "loading";
const tv_13 = data.data["sensor.rockuenergye3_energy_voltage_2_2"] ? data.data["sensor.rockuenergye3_energy_voltage_2_2"].state : "loading";
const ti_11 = data.data["sensor.rockuenergye3_energy_current_0_2"] ? data.data["sensor.rockuenergye3_energy_current_0_2"].state : "loading";
const ti_12 = data.data["sensor.rockuenergye3_energy_current_1_2"] ? data.data["sensor.rockuenergye3_energy_current_1_2"].state : "loading";
const ti_13 = data.data["sensor.rockuenergye3_energy_current_2_2"] ? data.data["sensor.rockuenergye3_energy_current_2_2"].state : "loading";
const pp_11 = data.data["sensor.rockuenergye3_energy_power_0"] ? data.data["sensor.rockuenergye3_energy_power_0"].state : "loading";
const pp_12 = data.data["sensor.rockuenergye3_energy_power_1"] ? data.data["sensor.rockuenergye3_energy_power_1"].state : "loading";
const pp_13 = data.data["sensor.rockuenergye3_energy_power_2"] ? data.data["sensor.rockuenergye3_energy_power_2"].state : "loading";
const pi_11 = data.data["sensor.rockuenergye3_energy_current_0"] ? data.data["sensor.rockuenergye3_energy_current_0"].state : "loading";
const pi_12 = data.data["sensor.rockuenergye3_energy_current_1"] ? data.data["sensor.rockuenergye3_energy_current_1"].state : "loading";
const pi_13 = data.data["sensor.rockuenergye3_energy_current_2"] ? data.data["sensor.rockuenergye3_energy_current_2"].state : "loading";
const pv_11 = data.data["sensor.rockuenergye3_energy_voltage_0"] ? data.data["sensor.rockuenergye3_energy_voltage_0"].state : "loading";
const pv_12 = data.data["sensor.rockuenergye3_energy_voltage_1"] ? data.data["sensor.rockuenergye3_energy_voltage_1"].state : "loading";
const pv_13 = data.data["sensor.rockuenergye3_energy_voltage_2"] ? data.data["sensor.rockuenergye3_energy_voltage_2"].state : "loading";
const lc_01 = data.data["sensor.led_1_current"] ? data.data["sensor.led_1_current"].state : "loading";
const lc_02 = data.data["sensor.led_2_current"] ? data.data["sensor.led_2_current"].state : "loading";
const lc_03 = data.data["sensor.led_3_current"] ? data.data["sensor.led_3_current"].state : "loading";
const lv_01 = data.data["sensor.led_1_voltage"] ? data.data["sensor.led_1_voltage"].state : "loading";
const lv_02 = data.data["sensor.led_2_voltage"] ? data.data["sensor.led_2_voltage"].state : "loading";
const lv_03 = data.data["sensor.led_3_voltage"] ? data.data["sensor.led_3_voltage"].state : "loading";
const lp_01 = data.data["sensor.led_1_power"] ? data.data["sensor.led_1_power"].state : "loading";
const lp_02 = data.data["sensor.led_2_power"] ? data.data["sensor.led_2_power"].state : "loading";
const lp_03 = data.data["sensor.led_3_power"] ? data.data["sensor.led_3_power"].state : "loading";

```

รูปที่ 3.37 สร้างตัวแปรเก็บข้อมูล

10.) ทำการใช้ React และ JSX เพื่อดึงค่าจากตัวแปรต่าง ๆ แล้วนำมาแสดงบนหน้าจอ พร้อมทั้งใช้Tailwind CSS เพื่อจัดรูปแบบและตกแต่งการแสดงผล โดยมีการทำงานดังนี้:ในโค้ดนี้มีการแสดงค่าต่าง ๆ เช่น ความเข้มแสง (INTENSITY), ความชื้น (HUMIDITY), อุณหภูมิ (TEMP), ความเร็วลม (WINDSPEED) โดยการดึงค่าจากตัวแปร (เช่น li_out, hu_outdoor, tmp_outdoor, wi_out) มาแสดงบนหน้าจอ ค่าตัวแปรเหล่านี้ถูกใส่เข้าไปใน JSX ผ่านการใช้งานลีบ {} เพื่อทำให้ค่าตัวแปรนั้นถูกแสดงผลในส่วนที่ต้องการการจัดตำแหน่งและออกแบบเลย์เอาท์:โค้ดนี้ใช้ Tailwind CSS ในการออกแบบตำแหน่งและตกแต่ง เช่น การใช้คลาส “absolute, top-[%], left-[%], text-[vw],” และ flex-col เพื่อจัดการตำแหน่งขององค์ประกอบต่าง ๆ บนหน้าจอการใช้ z-[2] ช่วยในการจัดลำดับการซ้อนทับขององค์ประกอบ เช่น ข้อความที่อยู่บนรูปภาพพื้นหลัง () ค่าที่แสดงผล:ค่าต่าง ๆ เช่น INTENSITY (li_out), HUMIDITY (hu_outdoor), TEMP (tmp_outdoor), และ WIND SPEED (wi_out) จะแสดงพร้อมกับหน่วยที่เหมาะสม เช่น lux, %, °C, km/h นอกจากนี้ยังแสดงค่าแรงดันไฟฟ้า (V), กระแสไฟฟ้า (A) และกำลังไฟฟ้า (W/kW/h) สำหรับแต่ละเฟส (L1, L2, L3) โดยดึงข้อมูลจากตัวแปรต่าง ๆ เช่น tv_L1, ti_L1, pv_L1, pi_L1 โค้ดนี้เป็นการดึงค่าต่าง ๆ จากตัวแปรที่เกี่ยวข้องกับข้อมูลสภาพแวดล้อม และข้อมูลไฟฟ้า เช่น อุณหภูมิ, แรงดันไฟฟ้า, กระแสไฟฟ้า และอื่น ๆ มาทำการแสดงผลบนหน้าจอ พร้อมกับใช้ Tailwind CSS ในการตกแต่งเพื่อให้การแสดงผลดูเป็นระเบียบและปรับขนาดได้ตามหน้าจอ

```
<div className="relative w-[100%]>
  
  <div className="absolute top-[7%] left-[4%] text-[1.7vw] z-[2] flex flex-col">
    <h1 className="text-black">INTENSITY : {li_out} lux</h1>
    <h1 className="text-black">HUMIDITY : {hu_outdoor} %</h1>
    <h1 className="text-black">TEMP : {tmp_outdoor} °C</h1>
    <h1 className="text-black">WIND SPEED : {wi_out} km/h</h1>
  </div>
  <div className="absolute top-[28%] left-[4%] text-[1.7vw] z-[2] flex flex-col">
    <h1 className="text-black">L1 : {tv_l1} V {ti_l1} A {tp_l1} W</h1>
    <h1 className="text-black">L2 : {tv_l2} V {ti_l2} A {tp_l2} W</h1>
    <h1 className="text-black">L3 : {tv_l3} V {ti_l3} A {tp_l3} W</h1>
  </div>
  <div className="absolute top-[47%] left-[4%] text-[1.7vw] z-[2] flex flex-col">
    <h1 className="text-black">L1 : {pv_l1} V {pi_l1} A {pp_l1} W</h1>
    <h1 className="text-black">L2 : {pv_l2} V {pi_l2} A {pp_l2} W</h1>
    <h1 className="text-black">L3 : {pv_l3} V {pi_l3} A {pp_l3} W</h1>
  </div>
  <div className="absolute top-[21%] left-[calc(45%+3px)] text-[1vw] z-[2] flex flex-col">
    <h1 className="text-black">TEMP : {tmp_06} °C</h1>
    <h1 className="text-black">TEMP : {(tmp_06 * 1.8) + 32} F</h1>
    <h1 className="text-black">HUMIDITY : {hu_06} %</h1>
  </div>
  <div className="absolute top-[69%] left-[calc(45%+2px)] text-[1vw] z-[2] flex flex-col">
    <h1 className="text-black">TEMP : {tmp_02} °C</h1>
    <h1 className="text-black">TEMP : {ToF(tmp_02)} F</h1>
    <h1 className="text-black">HUMIDITY : {hu_02} %</h1>
  </div>
</div>
```

รูปที่ 3.38 นำข้อมูลมาแสดงผล

11.) API documentation ของ Home Assistant ที่แสดงรายการ endpoints ที่รองรับทั้งการ GET และ POST ข้อมูล GET Requests: GET /api/: ดึงข้อมูลเกี่ยวกับระบบของ Home Assistant ทั้งหมด GET /api/config: ดึงข้อมูลการตั้งค่าปัจจุบันของ Home Assistant GET /api/events: ดึงรายการของ event ที่ถูกส่งไปยัง Home Assistant GET /api/services: ดึงรายการของ service ที่สามารถใช้ได้ใน Home Assistant GET /api/history/period/<timestamp>: ดึงข้อมูลประวัติ (history) ของ entities ในช่วงเวลาที่ระบุ GET /api/logbook/<timestamp>: ดึงข้อมูลจาก logbook ในช่วงเวลาที่ระบุ GET /api/states: ดึงสถานะของทุก entity ใน Home Assistant GET /api/states/<entity_id>: ดึงสถานะของ entity ที่ระบุ GET /api/error_log: ดึงบันทึกข้อผิดพลาด (error log) ของระบบ GET /api/camera_proxy/<camera entity_id>: ดึงข้อมูลภาพจากกล้อง (camera) ผ่าน entity ID ของกล้อง GET /api/calendars: ดึงรายการปฏิทินที่เชื่อมต่ออยู่กับ Home Assistant GET /api/calendars/<calendar entity_id>: ดึงข้อมูลของปฏิทินเฉพาะที่มี entity ID ที่ระบุ POST Requests: POST /api/states/<entity_id>: อัปเดตสถานะของ entity ที่ระบุ POST /api/events/<event_type>: และส่ง event ชนิดที่ระบุไปยัง Home Assistant POST /api/services/<domain>/<service>: เรียกใช้บริการใน domain ที่ระบุ POST /api/template: ส่ง template เพื่อรันคำสั่งแบบ jinja ใน Home Assistant POST /api/config/core/check_config: ตรวจสอบไฟล์ config ของ Home Assistant POST /api/intent/handle: จัดการคำสั่ง intent ที่ส่งไปยัง Home Assistant

Actions		
The API supports the following actions:		
GET	/api/	🔒
GET	/api/config	🔒
GET	/api/events	🔒
GET	/api/services	🔒
GET	/api/history/period/<timestamp>	🔒
GET	/api/logbook/<timestamp>	🔒
GET	/api/states	🔒
GET	/api/states/<entity_id>	🔒
GET	/api/error_log	🔒
GET	/api/camera_proxy/<camera entity_id>	🔒
GET	/api/calendars	🔒
GET	/api/calendars/<calendar entity_id>	🔒
POST	/api/states/<entity_id>	🔓
POST	/api/events/<event_type>	🔓
POST	/api/services/<domain>/<service>	🔓
POST	/api/template	🔓
POST	/api/config/core/check_config	🔓
POST	/api/intent/handle	🔓

รูปที่ 3.39 Home assistant API swagger document

12.) ดึงข้อมูลจาก Rest API ของ Home Assistant โดยใช้ Axios ใน React component ซึ่งทำงานโดยมีการกำหนดตัวแปร api ที่เก็บค่า IP และ key (token) ซึ่งดึงมาจากตัวแปรสิ่งแวดล้อม (environment variables) process.env.SOCKET_IP และ process.env.SERVER_TOKEN การตั้งค่า config สำหรับ headers เพื่อเพิ่ม Authorization แบบ Bearer token สำหรับการยืนยันตัวตน (Bearer \${api.key}) การใช้ useState มีการประกาศ state สองตัว: data และ data2 โดย setData และ setData2 ใช้สำหรับเก็บค่าข้อมูลที่ดึงมาจากการใช้ useEffect จะทำการเรียกฟังก์ชัน getdata() และ getdata2() เมื่อ component ถูก mount (ทำงานครั้งแรก) ฟังก์ชัน getdata(): เป็นฟังก์ชันแบบ async ที่ใช้ Axios ในการส่งคำขอ (request) แบบ GET ไปที่ endpoint /api/states ของ Home Assistant จากนั้นข้อมูลที่ได้ (dose.data) จะถูกเก็บไว้ใน state data ด้วย setData(dose.data) ฟังก์ชัน getdata2(): คล้ายกับ getdata() แต่จะส่งคำขอไปที่ endpoint /api/config เพื่อดึงข้อมูลการตั้งค่าของ Home Assistant และเก็บข้อมูลนั้นไว้ใน state data2 ด้วย setData2(dose.data)

```
import axios from "axios";

function Info() {
  const api = {
    IP: process.env.SOCKET_IP,
    key: process.env.SERVER_TOKEN,
  };
  const config = {
    headers: { Authorization: `Bearer ${api.key}` },
  };
  const [data, setData] = useState<any>([]);
  const [data2, setData2] = useState<any>([]);
  useEffect(() => {
    async function getdata() {
      const dose = await axios.get(
        `${api.IP}/api/states`,
        config,
      );
      console.log(dose.data);
      setData(dose.data);
    }
    async function getdata2() {
      const dose = await axios.get(
        `${api.IP}/api/config`,
        config,
      );
      setData2(dose.data);
    }
    getdata2();
    getdata();
  }, []);
}
```

รูปที่ 3.40 การดึงข้อมูลจาก Home assistant แบบ Restful API

13.) รันคำสั่ง pnpm dev เพื่อเริ่มต้นโหมดการพัฒนา (development mode) ของโปรเจกต์ React ที่ใช้ Vite เป็นเครื่องมือสำหรับการบันดิล (bundling) โดยมีการทำงานดังนี้: คำสั่ง pnpm dev เป็นคำสั่งที่บอก pnpm ให้เรียกสคริปต์ dev ที่กำหนดไว้ในไฟล์ package.json ของโปรเจกต์ (ในที่นี้เป็นการรัน Vite) คำสั่งนี้จะเปิดเซิร์ฟเวอร์ท้องถิ่น (local development server) เพื่อให้สามารถพัฒนาและทดสอบโปรเจกต์ React ในเบราว์เซอร์ รายละเอียดในผลลัพธ์: ตรวจสอบพอร์ต (Port)

โครงการนี้พยายามเปิดเซิร์ฟเวอร์ที่พอร์ต 5173 แต่เนื่องจากพอร์ตนี้ถูกใช้งานอยู่ มันจึงพยายามหาพอร์ตอื่นไป (5174, 5175) จนกระทั่งหาพอร์ตที่ว่าง ซึ่งในที่สุดเปิดได้ที่พอร์ต 5176

```
parad@dose MINGW64 ~/Downloads/react-shadcn-ui-template-main (main)
$ pnpm dev

> react-shadcn-ui-template@1.0.0 dev C:\Users\parad\Downloads\react-shadcn-ui-template-main
> vite

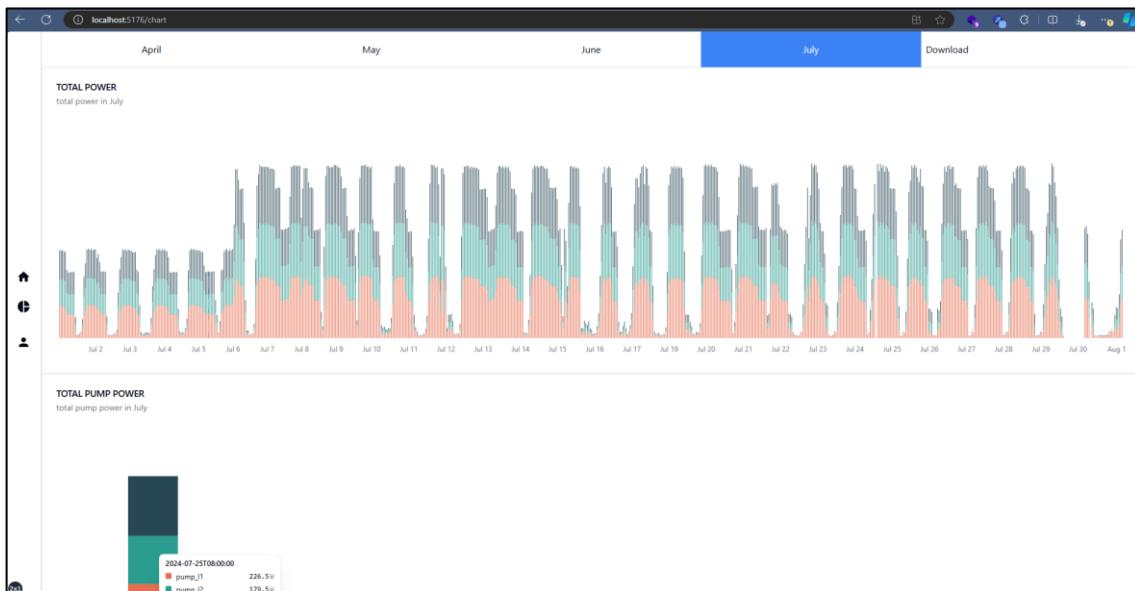
Port 5173 is in use, trying another one...
Port 5174 is in use, trying another one...
Port 5175 is in use, trying another one...

VITE v5.3.4 ready in 2477 ms

→ Local: http://localhost:5176/
→ Network: use --host to expose
→ press h + enter to show help
```

รูปที่ 3.41 รันโค้ดด้วย node package module

14.) เปิด Local บนบราวเซอร์: <http://localhost:5176> สำหรับทดสอบโครงการในเบราว์เซอร์ภายในเครื่อง มีการแนะนำให้ใช้ตัวเลือก --host เพื่อทำให้เซิร์ฟเวอร์เข้าถึงได้จากเครือข่ายภายนอก (เช่น อุปกรณ์อื่นในเครือข่ายเดียวกัน)



รูปที่ 3.42 หน้าแสดงผลบนบราวเซอร์

15.) ใช้ฟังก์ชัน defineConfig เพื่อกำหนดค่าการตั้งค่าต่างๆ ใน Vite โดยมีการตั้งค่า: plugins: [react()]: เปิดใช้งานปลั๊กอิน React ในโปรเจกต์ VitePWA: ใช้สำหรับตั้งค่า PWA โดยมีการกำหนดดังนี้: registerType: 'autoUpdate': ให้ PWA อัปเดตตัวเองโดยอัตโนมัติเมื่อมีการเปลี่ยนแปลง devOptions: { enabled: true }: เปิดใช้งานโหมดพัฒนา (development mode) สำหรับ PWA manifest: ตั้งค่าไฟล์ manifest ของ PWA name: ชื่อเต็มของแอปพลิเคชัน (ในที่นี้คือ "Greenhouse MJU") short_name:

ชื่อปุ่มของแอป (ใช้แสดงในหน้าจอหรือแท็บ) theme_color: สีธีมหลักของแอป (#14512e) start_url: URL ที่จะโหลดเมื่อเริ่มต้นแอป (ในที่นี้คือ "/") display: 'standalone': กำหนดให้แอปแสดงผลในโหมดสแตนด์โอลิน (ไม่แสดง UI ของเบราว์เซอร์) background_color: สีพื้นหลังของแอปขณะโหลด (เป็นสีเดียวกับ theme color) icons: ไอคอนที่ใช้ในแอป โดยมีการกำหนดขนาดและชนิดของไฟล์ไอคอนที่ใช้

```
vite.config.ts > default > plugins > manifest > icons
1 import path from "path";
2 import react from "@vitejs/plugin-react";
3 import { defineConfig } from "vite";
4 import {VitePWA} from 'vite-plugin-pwa'
5
6 // https://vitejs.dev/config/
7 export default defineConfig({
8   plugins: [react(),
9     VitePWA({
10       registerType: 'autoUpdate',
11       devOptions:{,
12         enabled:true
13       },
14       manifest: {
15         name:'Greenhouse MJU',
16         short_name: 'GH_MJU',
17         theme_color: '#14512e',
18         start_url: '/',
19         display: 'standalone',
20         background_color: '#14512e',
21         icons: [
22           {
23             src: '/logo.png',
24             sizes: '192x192',
25             type: 'image/png',
26           },
27           [
28             {
29               src: '/logo.png',
30               sizes: '512x512',
31               type: 'image/png',
32             },
33           ],
34         },
35       ],
36       resolve: {
37         alias: {
38           "@": path.resolve(__dirname, "./src"),
39         },
40       },
41     })
42   ],
43   resolve: {
44     alias: {
45       "@": path.resolve(__dirname, "./src"),
46     },
47   },
48 })
49 })
```

รูปที่ 3.43 เช็ต progressive web app

16.) ทำการ build โปรเจกต์โดยใช้คำสั่ง pnpm build ซึ่งคำสั่งนี้จะทำการคอมไพล์และบรรจุโค้ดทั้งหมดสำหรับการใช้งานจริง (production). ถือเป็นการเสร็จสิ้นขั้นตอน

```
C:\Users\parad\Greenhouse>pnpm build

> react@0.0.0 build C:\Users\parad\Greenhouse
> tsc -b && vite build

vite v5.4.9 building for production...
transforming (1) index.htmlDeprecation Warning: The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

✓ 105 modules transformed.
dist/index.html          0.46 kB | gzip:  0.30 kB
dist/assets/index-B6pQn35I.css 6.26 kB | gzip: 1.83 kB
dist/assets/index-qaN02Y-7.js 207.32 kB | gzip: 70.22 kB
✓ built in 4.47s
```

รูปที่ 3.44 การ build project ด้วย Vite

3.1.9 วิเคราะห์และทดสอบประสิทธิภาพ

ในส่วนนี้จะเป็นการวิเคราะห์ผลการใช้พลังงานในโรงเรือนโดยใช้ k6 load testing เพื่อทำการจำลองผู้ใช้งาน เพื่อส่งคำขอข้อมูลไปยัง IP Address ของระบบแสดงผลข้อมูล โดยมีโค้ดดังนี้ หลังจากรัน скриปต์สำหรับทดสอบระบบ จะทำให้ได้ผลการทดสอบประสิทธิภาพของระบบ อย่างละเอียดโดยมีการเขียนโค้ด ดังรูปที่ 3.45 code สำหรับทดสอบประสิทธิภาพ และจะได้ผลการทดสอบ ดังรูปที่ 3.46 ตัวอย่างผลการทดสอบประสิทธิภาพ

```
1 import http from 'k6/http';
2 import { sleep } from 'k6';
3
4 const IP = process.env.SERVER_IP;
5
6 export const options = {
7   vus: 3000,
8   duration: '30s',
9 }
10 export default function() {
11   http.get(IP);
12   sleep(1);
13 }
```

รูปที่ 3.45 code สำหรับทดสอบประสิทธิภาพ

```
PS C:\Users\parad\OneDrive\|\ ດែក ទី ៧ ចាប់ k6 run script.js

[ \ \ \ \ \ ] [ \ \ \ \ \ ] .io

execution: local
script: script.js
output: -


scenarios: (100.00%) 1 scenario, 3000 max VUs, 1m0s max duration (incl. graceful stop):
  * default: 3000 looping VUs for 30s (gracefulStop: 30s)

data_received.....: 41 MB 1.3 MB/s
data_sent.....: 3.0 MB 94 kB/s
http_req_blocked.....: avg=725.5ms min=0s med=0s max=17.69s p(90)=0s p(95)=6.84s
http_req_connecting.....: avg=418.05ms min=0s med=0s max=9.78s p(90)=0s p(95)=4.25s
http_req_duration.....: avg=721.36ms min=39.36ms med=433.74ms max=6.61s p(90)=1.65s p(95)=2.42s
{ expected_response:true }.....: avg=721.36ms min=39.36ms med=433.74ms max=6.61s p(90)=1.65s p(95)=2.42s
http_req_failed.....: 0.00% v 0 X 35944
http_req_receiving.....: avg=32.1ms min=0s med=0s max=2.28s p(90)=71.62ms p(95)=120.04ms
http_req_sending.....: avg=1.88ms min=0s med=0s max=2.72s p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=232.34ms min=0s med=0s max=7.64s p(90)=0s p(95)=1.92s
http_req_waiting.....: avg=687.37ms min=0s med=417.11ms max=6.31s p(90)=1.59s p(95)=2.23s
http_reqs.....: 35944 1122.469203/s
iteration_duration.....: avg=2.56s min=1.04s med=1.53s max=20.28s p(90)=4.5s p(95)=9.26s
iterations.....: 35944 1122.469203/s
vus.....: 1820 min=0 max=3000
vus_max.....: 3000 min=1334 max=3000
```

รูปที่ 3.46 ตัวอย่างผลการทดสอบประสิทธิภาพ

3.1.10 การบันทึกและวิเคราะห์ผล

นำข้อมูลที่ได้มาจากการบันทึกผ่าน Home Assistant มาจัดเรียงข้อมูล และเปรียบเทียบปริมาณการใช้พลังงานงาน และวิเคราะห์เปรียบเทียบอุณหภูมิภายในและภายนอกสร้างกราฟเปรียบเทียบอุณหภูมิภายในและภายนอกเพื่อดูความสัมพันธ์วิเคราะห์ปัจจัยที่มีผลต่ออุณหภูมิศึกษาปัจจัยต่างๆ ที่มีผลต่ออุณหภูมิภายใน เช่น การทำงานของระบบปรับอากาศ การเปิด-ปิดประตูหน้าต่าง และปริมาณแสงแดดทางสาเหตุของความแตกต่าง หาสาเหตุของความแตกต่างของอุณหภูมิในแต่ละจุดภายในโรงเรือน

3.1.11 สรุปผลและจัดทำรายงานผลโครงการ

สรุปผลการทดลองที่ได้ การวิเคราะห์ผล ประโยชน์จะได้รับ และการเปรียบเทียบกับงานวิจัยอื่นๆ ส่วนการจัดทำรายงานตามรูปแบบ รายงานของวิทยาลัยพลังงานทดแทน มหาวิทยาลัยแม่โจ้ ซึ่งจะประกอบด้วยส่วนต่างๆ ที่สำคัญ เช่น บทนำ วิธีดำเนินการ ผลการทดลอง และสรุปผล ซัดเจนและกระชับ ประกอบด้วยตาราง กราฟ และผลการวิเคราะห์ทางพลังงาน

3.2 แผนการดำเนินงาน

ระยะเวลาดำเนินงานวิจัยการศึกษา และติดตั้งระบบตรวจวัดและควบคุมโรงเรือนโดยใช้ระบบ IoT จะใช้เวลาประมาณ 5 เดือน โดยมีแผนการดำเนินงาน แสดงดังตารางที่ 3.2

ตารางที่ 3.1 แผนการดำเนินงาน

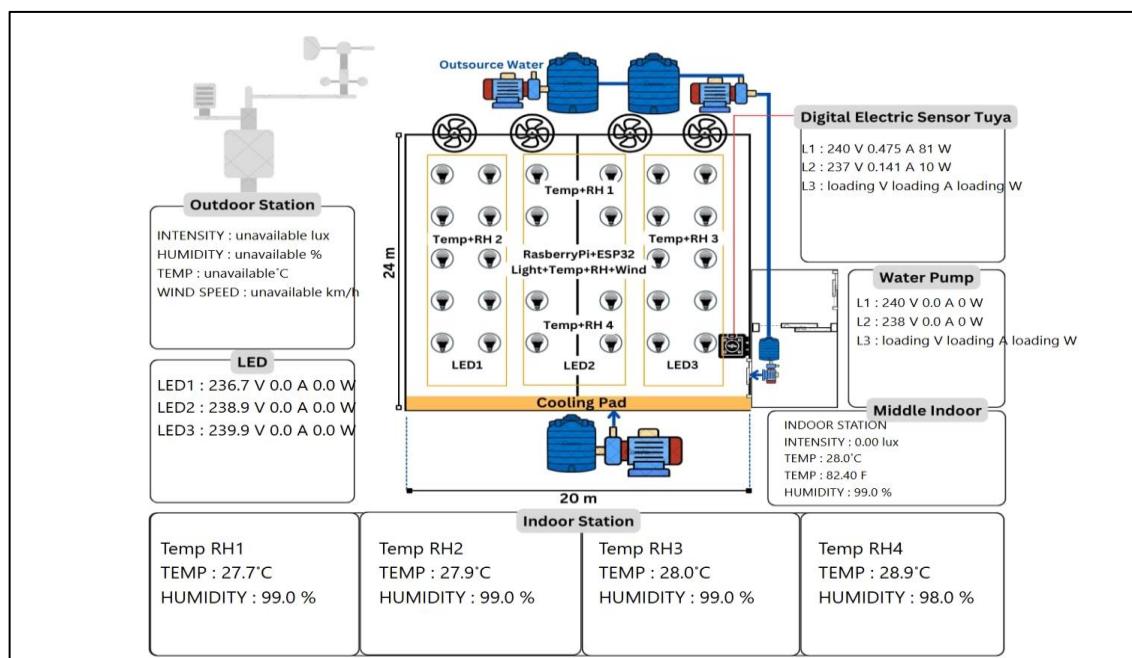
ขั้นตอนการดำเนินงาน	แผนการดำเนินงาน (เดือนที่)				
	1	2	3	4	5
ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง					
สำรวจอุปกรณ์จุดตรวจภายในโรงเรือน และจำลองแผนผัง					
จำลองแผนผังการวางแผนอุปกรณ์					
ออกแบบจุดติดตั้งเซ็นเซอร์					
ติดตั้ง Home Assistant Server					
ทำการออกแบบหน้าแสดงผลข้อมูล					
ออกแบบการรับและจัดเรียงข้อมูลเพื่อวิเคราะห์ผล					
บันทึกและวิเคราะห์ผล					
สรุปผลการดำเนินงานและจัดทำรายงานผลโครงการ					

บทที่ 4

ผลการทดลอง

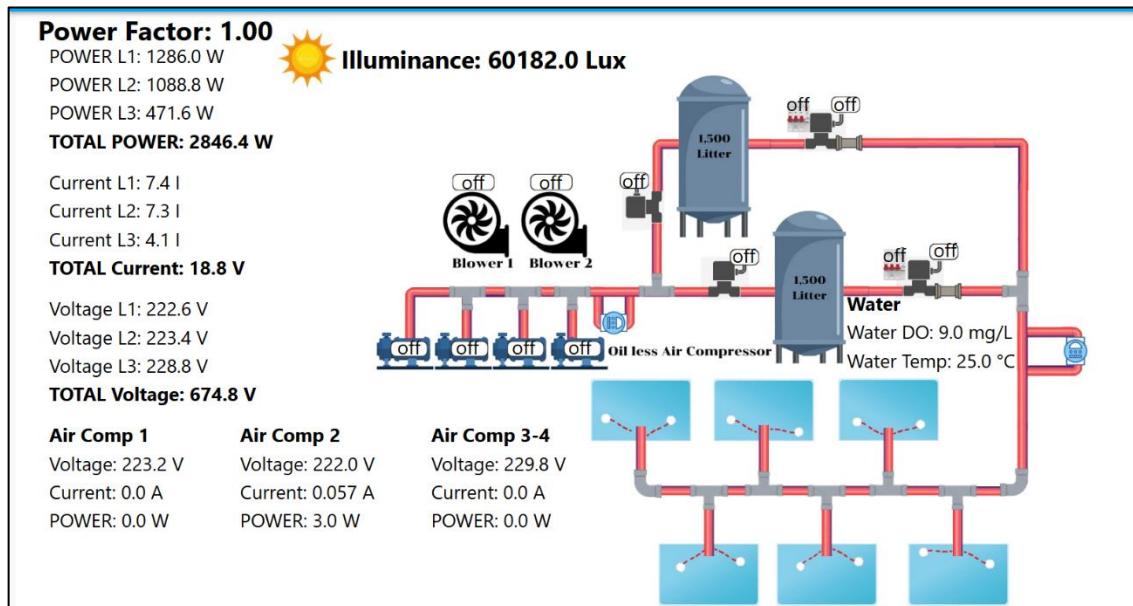
4.1 หน้าแสดงผลข้อมูล

การแสดงผลข้อมูลจากเซนเซอร์ต่างๆ ผ่าน Home Assistant โดยมีการรวมข้อมูลจากเซนเซอร์หลายประเภท มูลเหล่านี้ถูกรวบรวมและจัดการผ่าน Raspberry Pi หรือ ESP32 เพื่อให้แสดงผลใน Home Assistant ส่วนประกอบหลักในระบบจะมี Outdoor Station (สถานีก่อสร้าง) เซนเซอร์ตรวจจับสภาพแวดล้อมภายนอก เช่น ความเข้มแสง (INTENSITY), ความชื้น (HUMIDITY), อุณหภูมิ (TEMP), และความเร็วลม (WIND SPEED) LED Indicators แสดงค่าแรงดันไฟฟ้า (V), กระแสไฟฟ้า (A), และกำลังไฟฟ้า (W) ของ LED 3 ชุด Raspberry Pi ทำหน้าที่เป็นตัวกลางในการรวบรวมข้อมูลจากเซนเซอร์หลายตัว เช่น เซนเซอร์วัดอุณหภูมิ (Temp) และความชื้นสัมพัทธ์ (RH) ในสถานที่ต่างๆ รับข้อมูลจากเซนเซอร์สำหรับอุณหภูมิ ความชื้น แสง และลม เพื่อนำไปแสดงผลในระบบ Home Assistant Indoor Station (สถานีภายใน) มีการติดตั้งเซนเซอร์วัดอุณหภูมิและความชื้นในแต่ละตำแหน่งของพื้นที่ภายในโดยมีการแบ่งเป็น RH1 ถึง RH4 สำหรับการติดตามสภาพอากาศภายใน มีข้อมูลเพิ่มเติมที่แสดงอุณหภูมิและความชื้นในแต่ละจุด โดยบอกเป็นอุณหภูมิทั้งในหน่วยเซลเซียส (C) ปั๊มน้ำ (Water Pump) แสดงสถานะการทำงานของปั๊มและข้อมูลแรงดันไฟฟ้า (V), กระแสไฟฟ้า (A), และกำลังไฟฟ้า (W) Digital Electric Sensor Tuya ตรวจสอบการใช้พลังงานไฟฟ้าในระบบ และแสดงข้อมูลของแรงดันไฟฟ้า (V), กระแสไฟฟ้า (A), และพลังงานไฟฟ้าที่ใช้ (W) ในสามเฟส (L1, L2, L3) Middle Indoor Station แสดงอุณหภูมิกายในจุดกลางในระบบ หน่วยเป็นเซลเซียส รวมถึงค่าความชื้น



รูปที่ 4.1 หน้าแสดงผลข้อมูลของโรงเรือนกัญชา

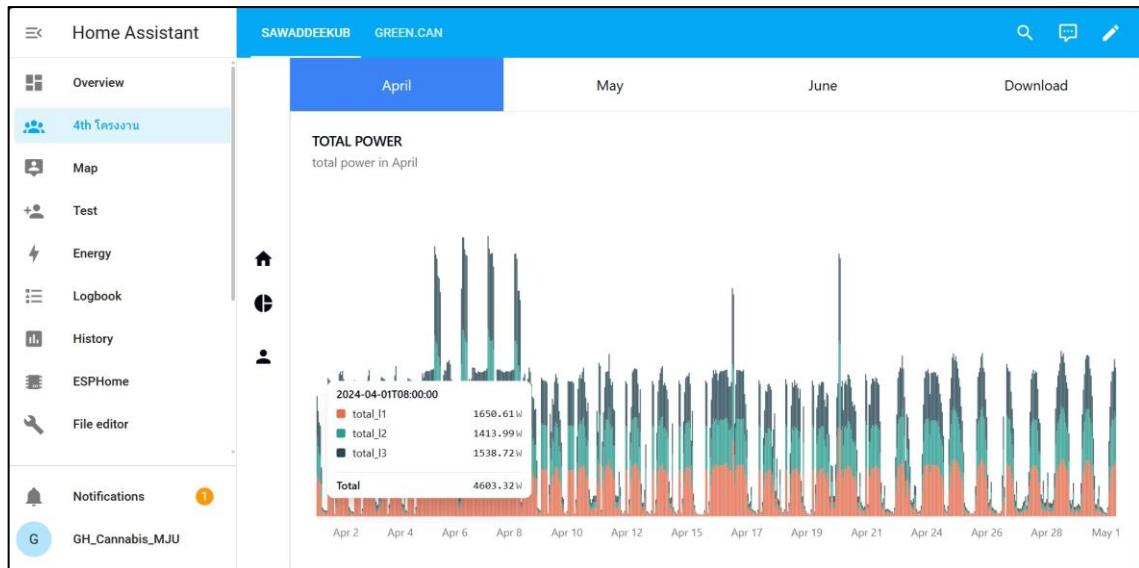
การจ่ายพลังงานในโรงเรือนบ่อกุ้งซึ่งประกอบด้วย Power Factor และค่าประสิทธิภาพการใช้พลังงานไฟฟ้าของระบบ Illuminance ระดับความสว่างในพื้นที่ Total Power Consumption และกำลังไฟฟ้าที่ใช้ในแต่ละเฟสและกำลังไฟฟ้ารวมของระบบ Current กระแสในแต่ละเฟสและกระแสรวมของระบบ Voltage แรงดันในแต่ละเฟสและแรงดันรวมของระบบ Air Compressor และค่าแรงดันและกระแสไฟฟ้าของเครื่องอัดอากาศแต่ละตัว Blower มีตัวเป่าลมที่สามารถเปิดหรือปิดการทำงานได้ Water System และค่าความเข้มข้นของออกซิเจนที่ละลายน้ำ (DO) และอุณหภูมิของน้ำ Tanks ถังเก็บน้ำ และท่อส่งน้ำที่เชื่อมต่อกัน



รูปที่ 4.2 หน้าแสดงผลของบ่อกุ้ง

4.1.1 Home assistant UI

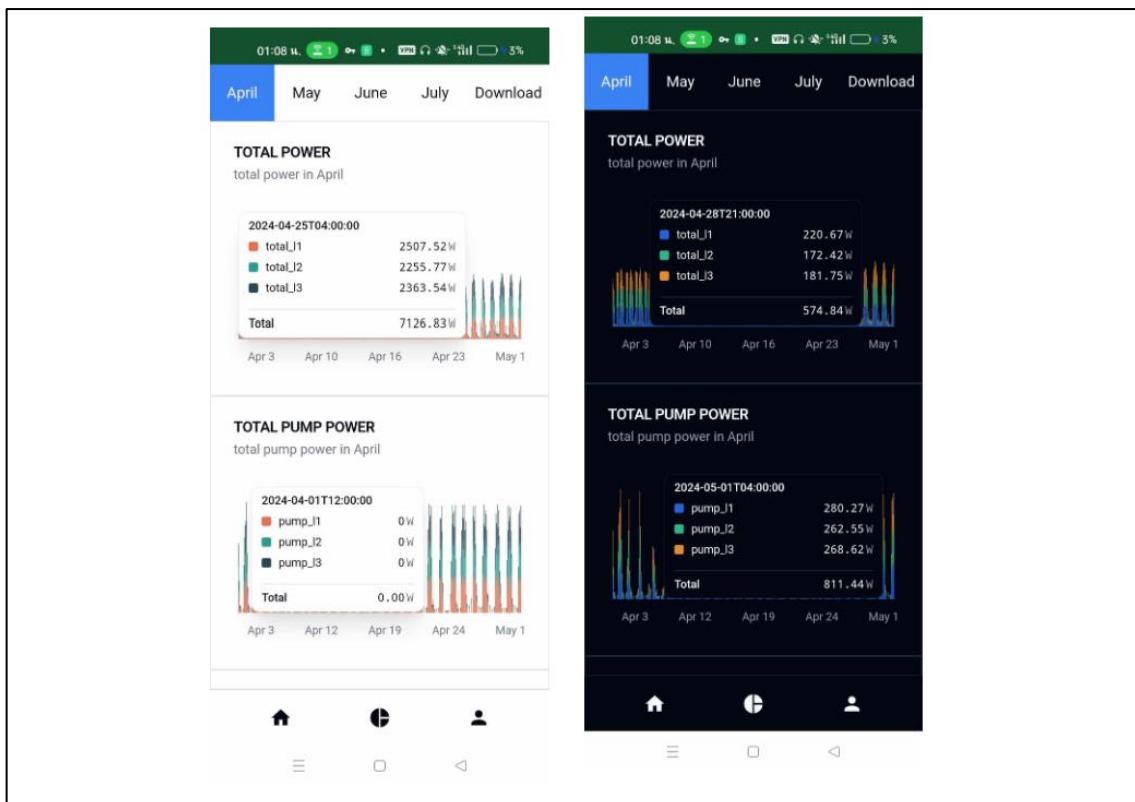
การนำ WebView มาใช้ใน Home Assistant (HA) เป็นการแสดงหน้าเว็บภายในแอปพลิเคชัน หรืออินเตอร์เฟซของ Home Assistant โดยใช้ฟังก์ชันที่ฝังหน้าเว็บให้ผู้ใช้สามารถดูข้อมูลจากเว็บไซต์ ภายนอกได้โดยไม่ต้องออกจากแอป หรือสามารถฝัง UI จากส่วนอื่น ๆ เข้ามาใน Home Assistant เพื่อเพิ่มฟังก์ชันการใช้งานเพิ่มเติมใช้ iFrame Card ใน Home Assistant รองรับการฝังหน้าเว็บโดยใช้ iFrame Card ซึ่งเป็นวิธีที่ง่ายที่สุดในการแสดงหน้าเว็บจากแหล่งภายนอกหรือภายในระบบในเดชบอร์ด ของ Home Assistant หน้าเว็บจะถูกแสดงผลในเดชบอร์ดของ Home Assistant โดยผู้ใช้สามารถเลื่อนดูเนื้อหาจากเว็บนั้นได้โดยตรง การแสดงผล Web UI ผ่าน WebView ในแอป Home Assistant เมื่อใช้ WebView ในแอป Home Assistant (เช่น การทำ Progressive Web App หรือใช้อัปบันมือถือที่ใช้ WebView) เราสามารถใช้มันในการแสดงเนื้อหาของ UI ที่สร้างขึ้นจากเทคโนโลยีเว็บต่าง ๆ ได้ สำหรับนักพัฒนาที่ต้องการแสดง UI เพิ่มเติมจากระบบอื่น หรือฝังหน้าเว็บเฉพาะใน Home Assistant



รูปที่ 4.3 หน้าแสดงผลข้อมูลบน Home assistant

4.1.2 Mobile

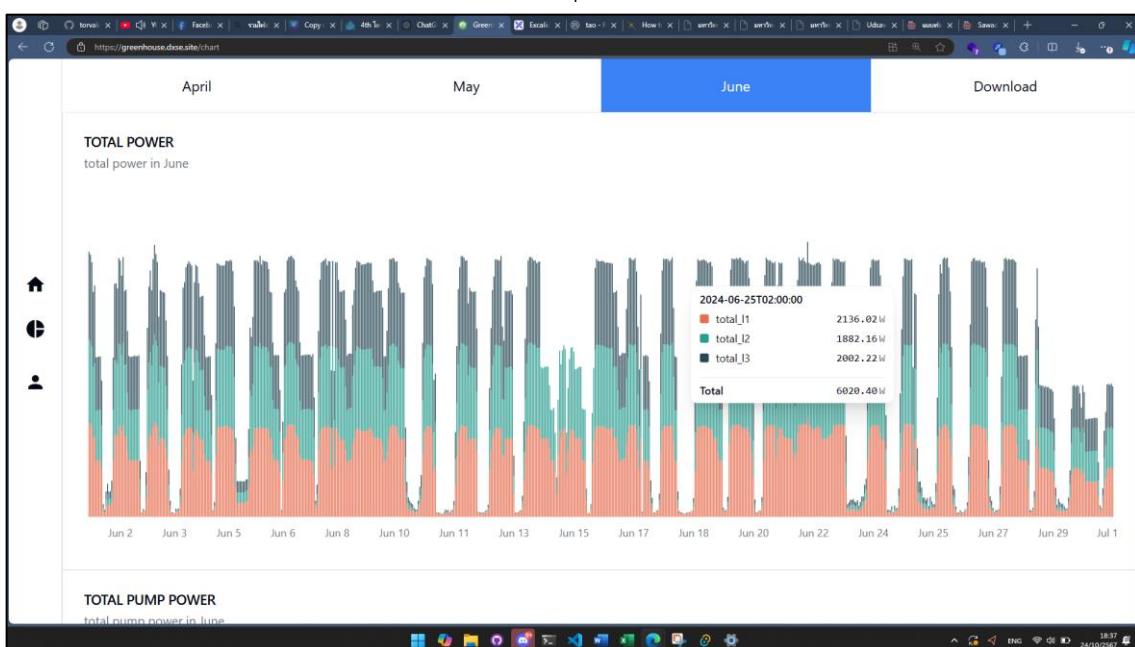
การแสดงผลของ Progressive Web App (PWA) ที่ถูกนำมาใช้แสดงข้อมูลการใช้พลังงานผ่านมือถือในระบบ Home Assistant ซึ่ง PWA คือเทคโนโลยีที่ช่วยให้อแอปพลิเคชันที่พัฒนาเป็นเว็บแอปสามารถทำงานได้คล้ายกับแอปพลิเคชันบนมือถือ ทำให้ผู้ใช้สามารถเข้าถึงข้อมูลหรือบริการได้อย่างสะดวกผ่านเบราว์เซอร์บนมือถือสองโหมดที่แตกต่างกัน คือ โหมดสว่าง และโหมดมืด ซึ่งแสดงถึงความสามารถของ PWA ในการปรับแต่ง UI ให้เข้ากับโหมดแสดงผลของระบบปฏิบัติการของมือถือโหมดมืด (Dark Mode) ช่วยลดแสงจ้าและถนอมสายตา โดยจะเปลี่ยนพื้นหลังเป็นสีเข้มและข้อความหรือกราฟเป็นสีสว่าง Progressive Web App (PWA) ช่วยให้ผู้ใช้สามารถเข้าถึง Home Assistant ได้อย่างสะดวกและมีประสิทธิภาพผ่านมือถือโดยไม่จำเป็นต้องติดตั้งแอปพลิเคชันแยกต่างหาก นอกจากนี้ยังมีคุณสมบัติเพิ่มเติมเช่น การทำงานในโหมดอффไลน์ การรองรับการแจ้งเตือน และความสามารถในการปรับ UI ให้เข้ากับโหมดการแสดงผลของระบบ



รูปที่ 4.4 หน้าแสดงผลบนมือถือ

4.1.3 Website

หน้าแสดงผลสำหรับ Home Assistant หรือการแสดงผลข้อมูลอื่น ๆ บนเว็บด้วย React.js และ Tailwind CSS เป็นกระบวนการที่ทำให้เว็บแอปพลิเคชันสามารถดึงข้อมูลจาก API และแสดงผลข้อมูลเหล่านั้นอย่างสวยงาม และใช้งานได้ดีในหลากหลายอุปกรณ์ (Responsive Design)



รูปที่ 4.5 หน้าแสดงผลบนบราวเซอร์

4.2 เช่นเชอร์ในHome assistant

หลังจากเพิ่มเช่นเชอร์ Tuya ใน Home Assistant แล้ว ผู้ใช้จะสามารถตรวจสอบสถานะของอุปกรณ์ต่าง ๆ แบบเรียลไทม์ได้ เช่น สถานะของไฟฟ้า การทำงานของบีบีน้ำ หรือระดับความสูงของแสง LED เป็นต้น ข้อมูลนี้จะแสดงในแดชบอร์ดของ Home Assistant ทำให้ผู้ใช้สามารถเห็นข้อมูลที่อัปเดตตลอดเวลา ดังภาพที่ รูปที่ 4.6 Tuya sensor

Device	Manufacturer	Model	Area	Integration	Battery
Breaker heat pump	Tuya	Smart Breaker	—	Tuya	—
Compteur Digital Electric 6	Tuya	Compteur Digital Electric	—	Tuya	—
L1-HP	Tuya	Compteur Digital Electric	—	Tuya	—
L2-HP	Tuya	Compteur Digital Electric	—	Tuya	—
LED-1	Tuya	Compteur Digital Electric	—	Tuya	—
LED-2	Tuya	Compteur Digital Electric	—	Tuya	—
LED-3	Tuya	Compteur Digital Electric	—	Tuya	—
P2 hot water pump	Tuya	Smart Breaker Power Meter	—	Tuya	—
P4 เครื่องปั๊มน้ำ	Tuya	Smart Breaker Power Meter	—	Tuya	—
Power of pump-1	Tuya	Compteur Digital Electric	—	Tuya	—
Power of pump-2	Tuya	Compteur Digital Electric	—	Tuya	—
Power of pump-4	Tuya	Compteur Digital Electric	—	Tuya	—

รูปที่ 4.6 Tuya sensor

อุปกรณ์ที่เข้ามือต่อผ่าน ESPHome ซึ่งเป็นเฟรมเวิร์กสำหรับการสร้างอุปกรณ์ IoT โดยเฉพาะ การเพิ่มอุปกรณ์ผ่าน ESPHome ใน Home Assistant จะช่วยให้สามารถอนิเตอร์และควบคุมอุปกรณ์ ต่าง ๆ ดังรูปที่ 4.7 ESP Home sensor

Device	Manufacturer	Model	Area	Integration	Battery
RockuSense Weather Station Pro c1fb2c	LungRock	RockuSense	—	ESPHome	—
RockuSense Weather Station Pro c20ec4	LungRock	RockuSense	—	ESPHome	—

รูปที่ 4.7 ESP Home sensor

4.3 ข้อมูลสภาพแวดล้อมภายในโรงเรือนกัญชา

4.3.1 อุณหภูมิภายในโรงเรือนกัญชา

เมื่อพิจารณาเปรียบเทียบทั้ง 5 จุด จะเห็นได้ว่า จุดที่ 3 มีช่วงการเปลี่ยนแปลงของอุณหภูมิและค่าเบี่ยงเบนมาตรฐานที่สูงที่สุด ทำให้เป็นจุดที่มีความผันผวนมากที่สุด และอาจต้องระมัดระวังในการควบคุมอุณหภูมิ จุดที่ 4 และจุดที่ 5 มีค่าเบี่ยงเบนมาตรฐานที่ต่ำที่สุด ซึ่งบ่งชี้ถึงความเสถียรของอุณหภูมิที่ดีโดยเฉพาะจุดที่ 5 ซึ่งมีความผันผวนน้อยที่สุดและมีการควบคุมอุณหภูมิได้อย่างมีประสิทธิภาพ จุดที่ 1 และ 2 มีความใกล้เคียงกันในเรื่องค่าเฉลี่ยอุณหภูมิและการกระจายตัวของอุณหภูมิ แต่จุดที่ 2 มีความผันผวนมากกว่าจุดที่ 1 เล็กน้อย โดยสรุป การควบคุมอุณหภูมิในโรงเรือนควรให้ความสำคัญกับจุดที่มีความผันผวนสูง เช่น จุดที่ 3 ซึ่งมีช่วงการเปลี่ยนแปลงของอุณหภูมิกว้างที่สุด ส่วนจุดที่ 4 และ 5 มีความเสถียรมากกว่าและเหมาะสมกับการเพาะปลูกในสภาพแวดล้อมที่ต้องการความคงที่ของอุณหภูมิ



รูปที่ 4.8 กราฟอุณหภูมิภายในโรงเรือน

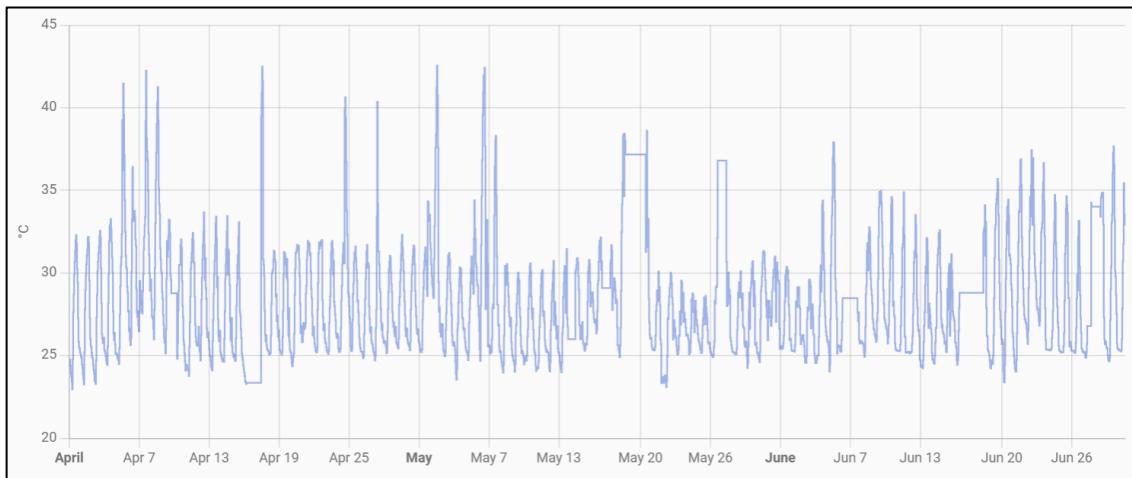
ตารางที่ 4.1 ข้อมูลอุณหภูมิในแต่ละจุดของโรงเรือน

อุณหภูมิจุดที่	ค่าต่ำสุด (°C)	ค่าสูงสุด (°C)	ค่าเฉลี่ย (°C)	ส่วนเบี่ยงเบนมาตรฐาน
1	22.97	42.54	28.20	3.34
2	22.54	43.10	28.06	3.44
3	22.91	44.01	28.37	3.58
4	22.12	41.56	27.03	3.17
5	23.07	42.26	27.68	3.06

4.3.1.1 อุณหภูมิจุดที่ 1

การเปลี่ยนแปลงของอุณหภูมิในโรงเรือนช่วงเดือนเมษายนถึงมิถุนายน มีค่าเฉลี่ยของอุณหภูมิ (AVG) อยู่ที่ 28.20°C ซึ่งเป็นอุณหภูมิที่เหมาะสมสำหรับการปลูกพืชหรือการดำเนินกิจกรรมในโรงเรือน เพราะไม่สูงหรือต่ำจนเกินไป ช่วยให้สภาพแวดล้อมมีเสถียรภาพ ซึ่งส่งผลดีต่อการเจริญเติบโตของพืชและ

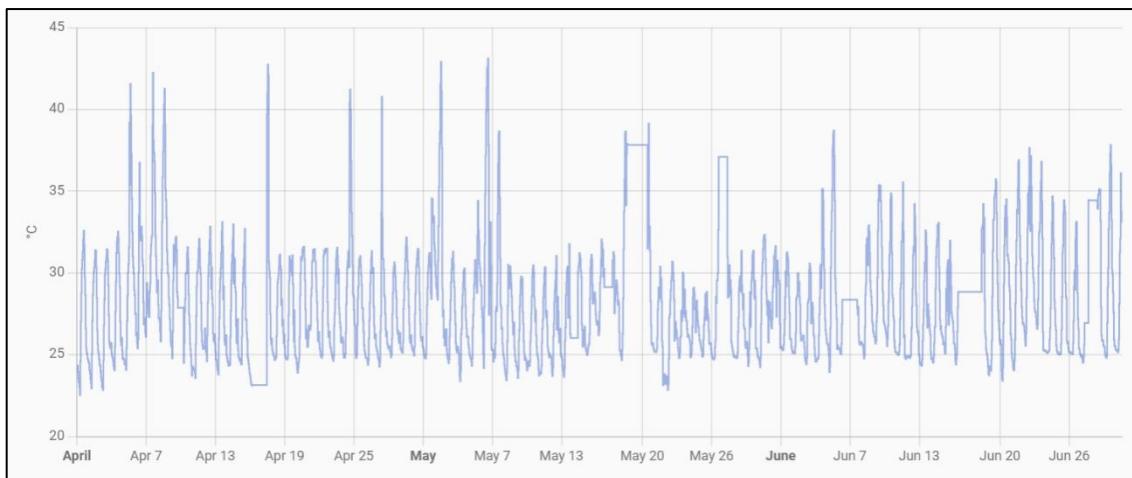
การจัดการในโรงเรือน อุณหภูมิที่ต่ำสุดในช่วงที่บันทึกได้คือ 22.97°C ซึ่งนักจะเกิดขึ้นในช่วงเวลากลางคืน ที่อุณหภูมิลดลง อุณหภูมิสูงสุดที่บันทึกได้คือ 42.54°C ซึ่งแสดงถึงความร้อนสะสมในช่วงกลางวันที่แสงแดดแรง ซึ่งในบางช่วงอาจส่งผลให้พืชเกิดความเครียดจากความร้อน ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) จากกราฟนี้ ค่าเบี่ยงเบนมาตรฐานอยู่ที่ 3.34 ซึ่งหมายความว่าอุณหภูมิที่บันทึกมีการผันผวนในระดับที่พอประมาณ แม้ว่าอุณหภูมิจะมีการเปลี่ยนแปลง แต่ค่าเบี่ยงเบนนี้ยังแสดงให้เห็นว่า อุณหภูมิส่วนใหญ่จะไม่เปลี่ยนแปลงไปไกลจากค่าเฉลี่ยมากนัก ทำให้ระบบการควบคุมอุณหภูมิยังคงเสถียรภาพในระดับหนึ่ง ดังรูปที่ 4.9 อุณหภูมิจุดที่ 1



รูปที่ 4.9 อุณหภูมิจุดที่ 1

4.3.1.2 อุณหภูมิจุดที่ 2

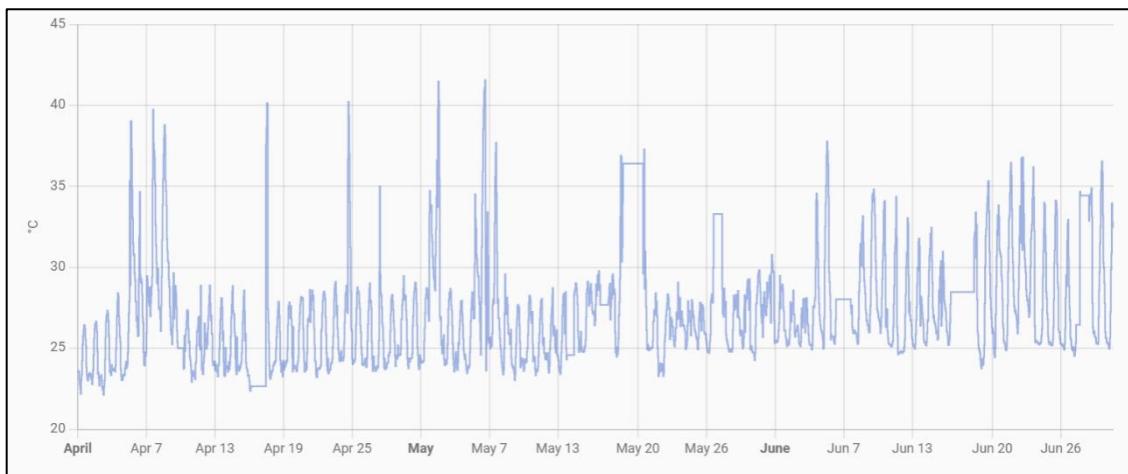
การเปลี่ยนแปลงของอุณหภูมิภายในจุดที่ 2 ของโรงเรือน โดยค่าเฉลี่ยอุณหภูมิที่ 28.06°C บ่งบอกถึงสภาพแวดล้อมที่ค่อนข้างอยู่ในช่วงอุณหภูมิปานกลางที่เหมาะสมสำหรับการเจริญเติบโตของพืชอย่างไรก็ตาม ความผันผวนของอุณหภูมิในจุดนี้อาจสังเกตได้จากค่าเบี่ยงเบนมาตรฐานที่ 3.44 ซึ่งแสดงว่า มีการกระจายตัวของอุณหภูมิจากค่าเฉลี่ยอยู่พอดีกับอุณหภูมิต่ำสุดที่ 22.54°C และสูงสุดที่ 43.10°C บ่งบอกถึงช่วงเวลาที่อุณหภูมิลดลงซึ่งอาจเกิดจากการควบคุมอุณหภูมิในเวลากลางคืนหรือการปรับระบายความร้อน ขณะที่ อุณหภูมิสูงสุดที่ 43.10°C บ่งบอกถึงช่วงเวลาที่โรงเรือนได้รับความร้อนสูง ซึ่งอาจเป็นผลจากสภาพแวดล้อมภายนอกหรือแสงแดดที่ร้อนแรงในช่วงเวลากลางวัน ดังรูปที่ 4.10 อุณหภูมิจุดที่ 2



รูปที่ 4.10 อุณหภูมิจุดที่ 2

4.3.1.3 อุณหภูมิจุดที่ 3

การเปลี่ยนแปลงของอุณหภูมิในจุดที่ 3 ภายในโรงเรือน โดยค่าเฉลี่ยที่ 28.37°C บ่งบอกถึงสภาพอุณหภูมิที่อยู่ในช่วงอุ่น ซึ่งเหมาะสมกับการเจริญเติบโตของพืชในโรงเรือน อย่างไรก็ตาม ค่าเบี่ยงเบนมาตรฐานที่ 3.58 แสดงถึงการกระจายตัวของอุณหภูมิที่มีความผันผวนอยู่บ้าง แต่ยังไม่ถือว่ามากเกินไป อุณหภูมิต่ำสุดที่ 22.91°C และสูงที่สุดที่ 44.01°C เป็นช่วงที่อุณหภูมิสูงขึ้นมาก ซึ่งอาจเกิดจากแสงแดดที่แรงหรือการระบายอากาศที่ดี ส่วนอุณหภูมิสูงสุดที่ 44.01°C เป็นช่วงที่อุณหภูมิสูงขึ้นมาก ซึ่งอาจเกิดจากแสงแดดที่แรงหรือการระบายความร้อนไม่เพียงพอในช่วงกลางวัน ดังรูปที่ 4.11 อุณหภูมิจุดที่ 3

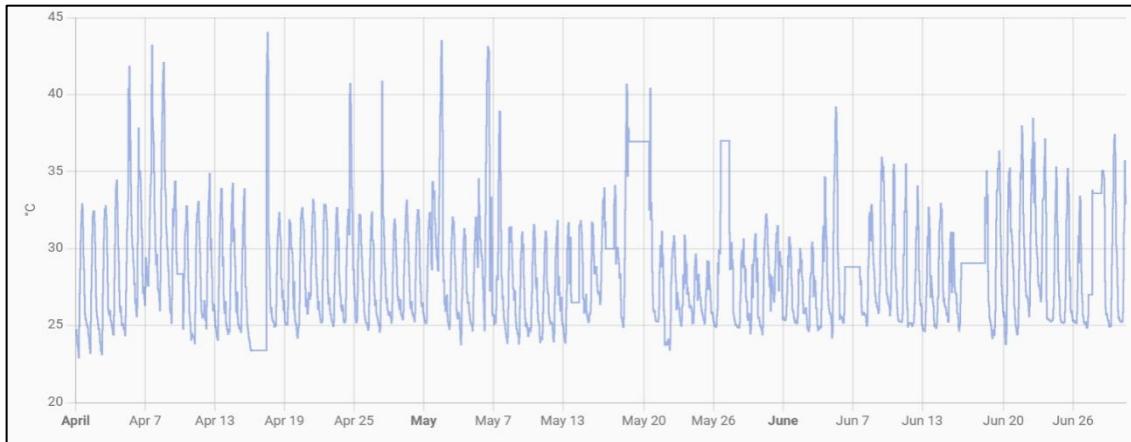


รูปที่ 4.11 อุณหภูมิจุดที่ 3

4.3.1.4 อุณหภูมิจุดที่ 4

การเปลี่ยนแปลงของอุณหภูมิกายในจุดที่ 4 โดยค่าเฉลี่ยของอุณหภูมิที่ 27.03°C บ่งบอกถึงสภาพอุณหภูมิที่ค่อนข้างอุ่นและเหมาะสมสำหรับการปลูกพืชในโรงเรือน ค่าเบี่ยงเบนมาตรฐานที่ 3.17 แสดงให้เห็นถึงการกระจายตัวของอุณหภูมิจากค่าเฉลี่ยอยู่ในระดับปานกลาง มีการเปลี่ยนแปลงของอุณหภูมิที่ไม่สูงมากนักในช่วงเวลาต่างๆ อุณหภูมิต่ำสุดที่ 22.12°C และสูงที่สุดที่ 41.56°C เป็นช่วงที่อุณหภูมิสูงสุดที่ 41.56°C ซึ่งอาจเกิดขึ้นในช่วงกลางคืนหรือเมื่อมีการระบายอากาศที่ดี ในขณะที่อุณหภูมิสูงสุดที่ 41.56°C

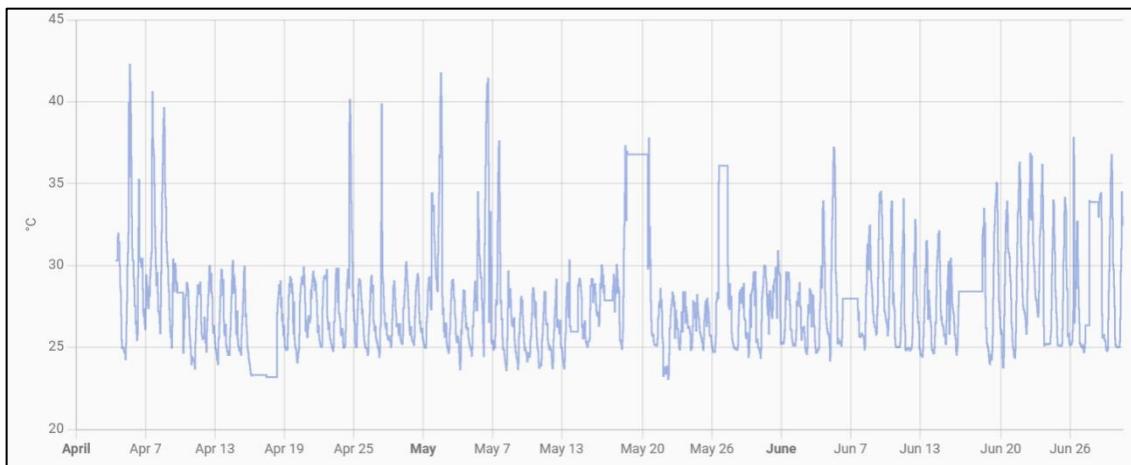
บ่งบอกถึงช่วงเวลาที่อุณหภูมิสูงมาก ซึ่งอาจเกิดจากแสงแดดหรืออุณหภูมิที่ค่อนข้างสูงเช่นเดียวกับจุดที่ 4 มีการเปลี่ยนแปลงของอุณหภูมิที่ค่อนข้างสม่ำเสมอ โดยมีค่าเบี่ยงเบนมาตรฐานที่ไม่สูงเกินไป ทำให้การเปลี่ยนแปลงของอุณหภูมิยังคงอยู่ในเกณฑ์ที่ควบคุมได้ อย่างไรก็ตาม อุณหภูมิที่สูงสุดที่ 41.56°C อาจต้องได้รับการระบายความร้อนเพิ่มเติมในช่วงกลางวันที่อากาศร้อนจัดเพื่อป้องกันการกระทบต่อการเจริญเติบโตของพืช ดังรูปที่ 4.12 อุณหภูมิจุดที่ 4



รูปที่ 4.12 อุณหภูมิจุดที่ 4

4.3.1.5 อุณหภูมิจุดที่ 5

การเปลี่ยนแปลงของอุณหภูมิในจุดที่ 5 โดยมีค่าเฉลี่ยที่ 27.68°C ซึ่งอยู่ในช่วงที่ค่อนข้างอุ่นและเหมาะสมสำหรับการเพาะปลูกหรือการใช้งานในโรงเรือน ค่าเบี่ยงเบนมาตรฐานที่ 3.06 บ่งบอกถึงการกระจายตัวของอุณหภูมิที่ไม่กว้างมาก ทำให้การเปลี่ยนแปลงของอุณหภูมิจากค่าเฉลี่ยไม่มากนัก ส่งผลให้สภาพแวดล้อมค่อนข้างเสถียร อุณหภูมิต่ำสุดที่ 23.07°C และสูงสุดที่ 42.26°C บ่งบอกถึงช่วงที่อากาศร้อนจัด ซึ่งอาจเป็นช่วงกลางวันที่แสงแดดแรง โดยรวม จุดที่ 5 มีการเปลี่ยนแปลงของอุณหภูมิที่ควบคุมได้ดี แม้ว่าจะมีช่วงอุณหภูมิสูงสุดที่ค่อนข้างร้อนที่ 42.26°C แต่ด้วยค่าเบี่ยงเบนมาตรฐานที่ต่ำ 3.06 บ่งชี้ว่าความผันผวนของอุณหภูมิไม่มาก ดังรูปที่ 4.13 อุณหภูมิจุดที่ 5



รูปที่ 4.13 อุณหภูมิจุดที่ 5

4.3.2 ความชื้นสัมพัทธ์ภายในโรงเรือนกัญชา

ค่าความชื้นสัมพัทธ์ต่ำสุดในแต่ละจุดมีช่วงระหว่าง 16.22% ถึง 27.91% ซึ่งแสดงให้เห็นว่าบางจุดความชื้นลดลงต่ำในบางช่วงเวลา จุดที่ 5 มีค่าต่ำสุดอยู่ที่ 16.22% ต่ำที่สุดในบรรดาจุดทั้งหมด แสดงถึงช่วงเวลาที่อากาศในจุดนั้นแห้งมากกว่าจุดอื่น ขณะที่จุดที่ 4 มีค่าต่ำสุดสูงที่สุดที่ 27.91% บ่งชี้ว่าความชื้นสัมพัทธ์ในจุดนั้นไม่ลดลงต่ำมากเหมือนจุดอื่น ๆ ซึ่งอาจมาจากปัจจัยภายนอกหรือระบบควบคุมความชื้นที่แตกต่างกัน ค่าสูงสุดของความชื้นสัมพัทธ์ในทุกจุดอยู่ที่ 99% ซึ่งสอดคล้องกับอย่างชัดเจน แสดงให้เห็นว่าโรงเรือนในทุกจุดอาจมีช่วงเวลาที่ความชื้นสัมพัทธ์พุ่งสูงมากในช่วงเวลาหนึ่ง แต่ไม่สามารถเกิน 100% ได้เนื่องจากเป็นขีดจำกัดของความชื้นสัมพัทธ์ การที่ค่าสูงสุดในทุกจุดอยู่ที่ 99% อาจสะท้อนถึงปัญหาหรือสถานการณ์ที่ทำให้ความชื้นในโรงเรือนขึ้นสูงจนใกล้ถึงจุดอิ่มตัว ซึ่งอาจเป็นผลจากสภาพอากาศภายนอกการจัดการโรงเรือน หรือระบบระบายอากาศที่ไม่เพียงพอ ค่าเฉลี่ยของความชื้นสัมพัทธ์ในแต่ละจุดอยู่ในช่วง 75.47% ถึง 82.99% ซึ่งหมายความว่าความชื้นในโรงเรือนโดยทั่วไปจะอยู่ในระดับค่อนข้างสูง จุดที่ 1 มีค่าเฉลี่ยสูงที่สุดอยู่ที่ 82.99% สะท้อนว่าพื้นที่นั้นมีความชื้นสูงโดยรวมเมื่อเทียบกับจุดอื่น ขณะที่จุดที่ 4 มีค่าเฉลี่ยต่ำที่สุดที่ 75.47% ความแตกต่างของค่าเฉลี่ยระหว่างจุดที่ 1 และจุดที่ 4 แสดงถึงความไม่สม่ำเสมอในการกระจายตัวของความชื้นภายในโรงเรือน อาจเนื่องจากระบบระบายอากาศที่ไม่สามารถกระจายความชื้นได้ทั่วถึงหรือการจัดการความชื้นในแต่ละจุดไม่เหมือนกัน ส่วนเบี่ยงเบนมาตรฐานของค่าความชื้นสัมพัทธ์ในแต่ละจุดบ่งบอกถึงความแปรผันของค่าความชื้นในแต่ละพื้นที่ โดยค่าที่สูงแสดงถึงการเปลี่ยนแปลงของความชื้นที่มากขึ้น จุดที่ 4 มีค่าความแปรผันสูงที่สุด 17.83 ซึ่งแสดงถึงการเปลี่ยนแปลงความชื้นในจุดนั้นมีความสม่ำเสมอมากกว่า ข้อมูลแสดงให้เห็นถึงความไม่สม่ำเสมอของความชื้นสัมพัทธ์ในโรงเรือน โดยเฉพาะในจุดที่ 4 ที่มีการแปรผันสูงกว่า นอกจากนั้น การที่ทุกจุดมีค่าสูงสุดของความชื้นอยู่ที่ 99% เป็นตัวบ่งชี้ถึงความชื้นในโรงเรือนที่อาจสูงเกินไปในบางช่วงเวลา ซึ่งเป็นเรื่องที่น่ากังวลในการจัดการโรงเรือน เนื่องจากความชื้นที่สูงมากเกินไปสามารถส่งผลกระทบต่อการเจริญเติบโตของพืชหรือสัตว์ในโรงเรือนได้



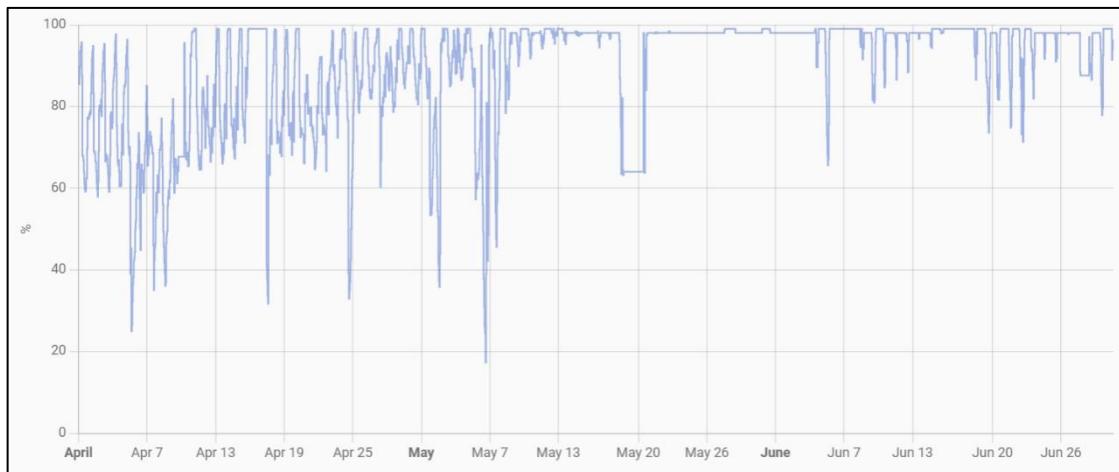
รูปที่ 4.14 กราฟความชื้นสัมพัทธ์ในโรงเรือนกัญชา

ตารางที่ 4.2 ข้อมูลความชื้นสัมพัทธ์ภายในโรงเรือน

ความชื้นสัมพัทธ์จุดที่	ค่าต่ำสุด (%)	ค่าสูงสุด (%)	ค่าเฉลี่ย (%)	ส่วนเบี่ยงเบนมาตรฐาน
1	17.36	99.00	82.99	15.35
2	20.57	99.00	82.84	15.12
3	20.26	99.00	82.25	15.22
4	27.91	99.00	75.47	17.83
5	16.22	99.00	82.77	15.39

4.3.2.1 ความชื้นสัมพัทธ์จุดที่ 1

ความชื้นสัมพัทธ์จุดที่ 1 นี้แสดงให้เห็นถึงระดับความชื้นสัมพัทธ์ที่มีการแพร่ผ่านค่อนข้างมาก โดยค่าเบี่ยงเบนมาตรฐานที่ 15.35 บ่งบอกถึงการกระจายตัวของข้อมูลความชื้นสัมพัทธ์จากค่าเฉลี่ย ซึ่งมีการแพร่ผ่านอย่างชัดเจนในแต่ละช่วงเวลา ค่าเฉลี่ยของความชื้นสัมพัทธ์อยู่ที่ 82.99% ซึ่งแสดงให้เห็นว่าสภาพแวดล้อมในจุดนี้มีความชื้นค่อนข้างสูง ค่าเฉลี่ยนี้บ่งบอกว่าสภาวะแวดล้อมโดยทั่วไปเหมาะสมสำหรับพืชหรือสัตว์ที่ต้องการความชื้นสูง แต่ก็ยังมีบางช่วงที่ความชื้นต่ำมากและบางช่วงที่ความชื้นสูงเกือบอิ่มตัว ส่วนค่าความชื้นสัมพัทธ์ต่ำสุดที่ 17.36% แสดงถึงช่วงเวลาที่ความชื้นลดลงอย่างมาก อาจเกิดจากการระบายอากาศที่ดีหรือสภาพอากาศภายนอกที่แห้ง ซึ่งส่งผลให้ความชื้นภายในโรงเรือนลดลงอย่างรวดเร็ว ในทางกลับกัน ความชื้นสัมพัทธ์สูงสุดที่บันทึกได้คือ 99% ซึ่งใกล้เคียงกับระดับอิ่มตัวของความชื้น (100%) อาจเกิดจากการปิดโรงเรือนเพื่อรักษาความชื้นหรือการที่ไม่มีการระบายอากาศเพียงพอในบางช่วงเวลา

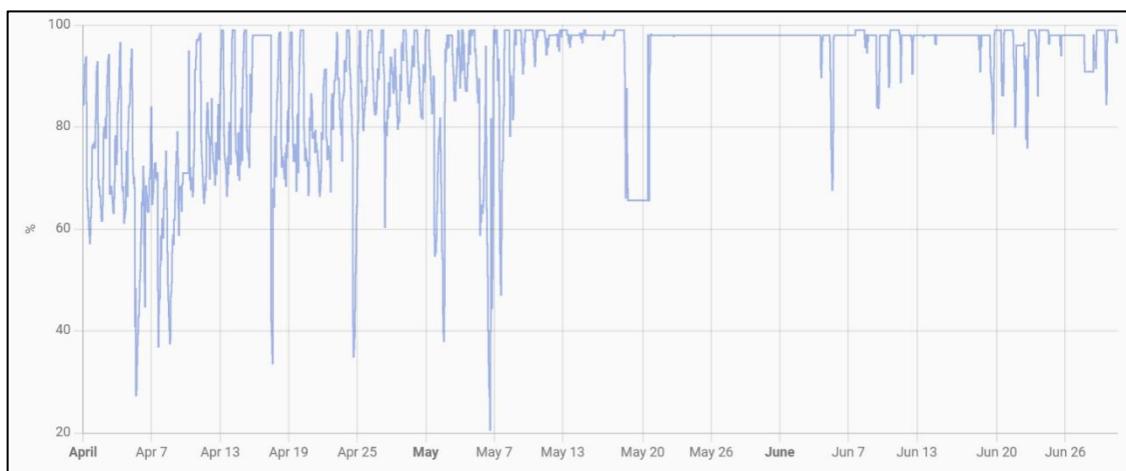


รูปที่ 4.15 ความชื้นสัมพัทธ์จุดที่ 1

4.3.2.2 ความชื้นสัมพัทธ์จุดที่ 2

ความชื้นสัมพัทธ์จุดที่ 2 นี้แสดงค่าความชื้นสัมพัทธ์ต่ำสุด 20.57% แสดงถึงช่วงเวลาที่ความชื้นลดลงมาก อาจเกิดจากการระบายอากาศที่ดีหรือสภาพอากาศภายนอกที่แห้ง เช่นเดียวกับจุดที่ 1 ซึ่งความชื้นต่ำนี้อาจไม่เหมาะสมกับพืชหรือสัตว์ที่ต้องการความชื้นสูง ค่าความชื้นสัมพัทธ์สูงสุด: 99.00% บ่งชี้ว่ามีบางช่วงเวลาที่ความชื้นเกือบถึงระดับอิ่มตัว ซึ่งอาจเกิดจากการปิดโรงเรือนเพื่อรักษาความชื้น

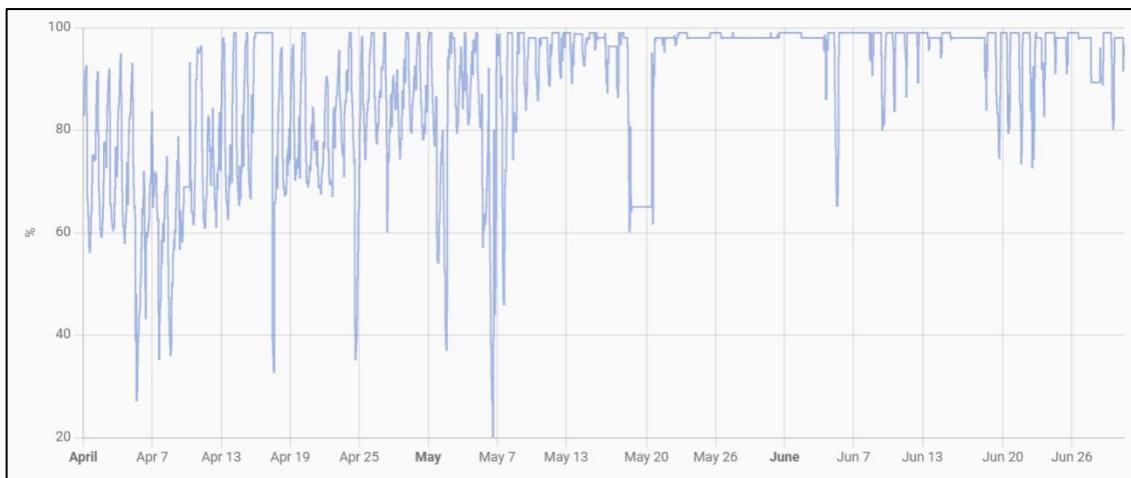
หรือไม่มีการระบายน้ำอากาศเพียงพอ ค่าเฉลี่ยของความชื้นสัมพัทธ์ 82.84% ใกล้เคียงกับจุดที่ 1 สะท้อนให้เห็นว่าสภาพแวดล้อมโดยรวมในจุดนี้มีความชื้นค่อนข้างสูง เหมาะสมสำหรับพืชหรือสัตว์บางประเภท แต่ก็ยังต้องระวังช่วงเวลาที่ความชื้นสูงเกินไป เป็นมาตรฐาน 15.12 แสดงให้เห็นว่ามีความผันผวนของความชื้นสัมพัทธ์ในระดับที่ใกล้เคียงกับจุดที่ 1 ซึ่งบ่งบอกว่าระบบควบคุมความชื้นยังคงมีการเปลี่ยนแปลงมากพอสมควร จุดที่ 2 มีลักษณะความชื้นสัมพัทธ์ที่ค่อนข้างคล้ายกับจุดที่ 1 โดยค่าเฉลี่ยของความชื้นอยู่ที่ประมาณ 82.84% ซึ่งเป็นระดับที่สูงเหมาะสมกับพืชบางชนิด อย่างไรก็ตาม ความชื้นที่แปรผันมากในแต่ละช่วงเวลาด้วยค่าเบี่ยงเบนมาตรฐานที่สูง 15.12 แสดงว่าระบบควบคุมอาจยังไม่เสถียร การปรับปรุงระบบการควบคุมความชื้นเพื่อให้เสถียรและคงที่มากขึ้นจะช่วยลดปัญหาที่เกิดจากความชื้นที่ผันผวน



รูปที่ 4.16 ความชื้นสัมพัทธ์จุดที่ 2

4.3.2.3 ความชื้นสัมพัทธ์จุดที่ 3

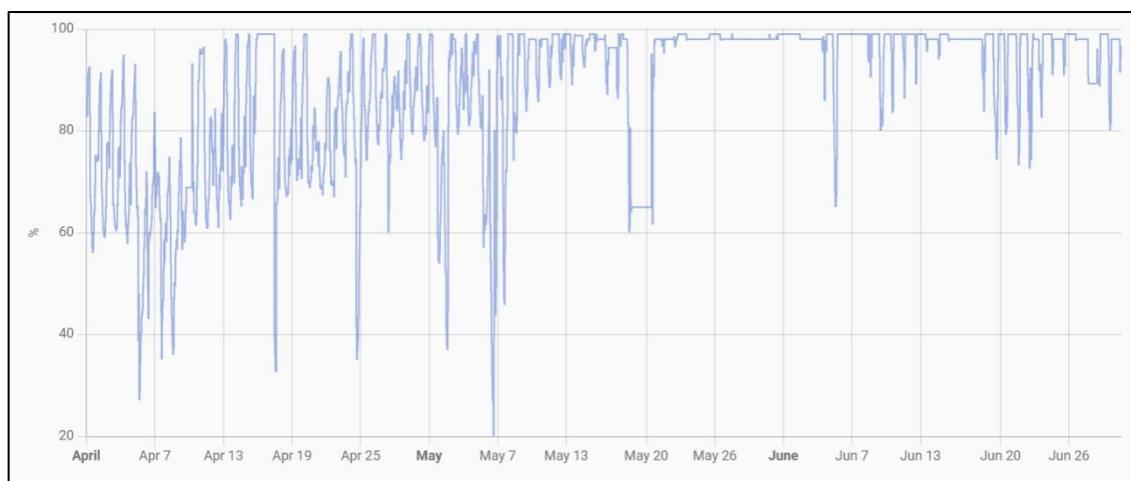
ค่าความชื้นสัมพัทธ์ต่ำสุด 20.26% แสดงถึงช่วงเวลาที่ความชื้นลดลงอย่างมาก ซึ่งอาจเกิดจาก การระบายน้ำอากาศที่มีประสิทธิภาพหรือสภาพอากาศภายนอกที่แห้งมาก การลดลงของความชื้นในระดับนี้อาจไม่เป็นผลดีต่อพืชที่ต้องการความชื้นสูง ค่าความชื้นสัมพัทธ์สูงสุด 99.00% แสดงให้เห็นถึงช่วงเวลาที่ความชื้นในโรงเรือนเกือบถึงจุดอิ่มตัว ซึ่งอาจเป็นผลมาจากการปิดโรงเรือนเพื่อรักษาความชื้นหรือมีการควบคุมการระบายน้ำอากาศที่ไม่เพียงพอ ความชื้นสูงระดับนี้อาจสร้างความเสี่ยงต่อการเกิดโรคจากเชื้อรา ค่าเฉลี่ยความชื้นสัมพัทธ์ 82.25% ซึ่งบ่งบอกถึงสภาพแวดล้อมที่มีความชื้นสูงต่อเนื่อง เหมาะสมสำหรับพืชที่ต้องการความชื้นสูง เช่น พืชในสภาพแวดล้อมชื้นแบบป่าเขตร้อน เป็นมาตรฐาน 15.22 บ่งชี้ถึงความผันผวนของค่าความชื้นที่ค่อนข้างสูง ซึ่งแสดงว่าระบบควบคุมความชื้นยังคงไม่เสถียรและต้องมีการปรับปรุงเพื่อรักษาความชื้นให้คงที่มากขึ้น



รูปที่ 4.17 ความชื้นสัมพัทธ์จุดที่ 3

4.3.2.4 ความชื้นสัมพัทธ์จุดที่ 4

ค่าความชื้นสัมพัทธ์ต่ำสุด 27.91% แสดงถึงช่วงเวลาที่ความชื้นลดลงอย่างมาก ซึ่งบ่งบอกถึงการระบายอากาศที่มีประสิทธิภาพหรือสภาพอากาศที่แห้งมากในบางช่วงเวลา การลดลงในระดับนี้อาจทำให้พืชที่ต้องการความชื้นสูงไม่สามารถเจริญเติบโตได้ดี ค่าความชื้นสัมพัทธ์สูงสุด 99.00% สะท้อนถึงช่วงเวลาที่ความชื้นเกือบถึงระดับอุ่นตัว การที่ความชื้นสูงถึงระดับนี้อาจเกิดจากการปิดโรงเรือนหรือระบบระบายอากาศที่ทำงานไม่เพียงพอ ซึ่งมีความเสี่ยงต่อการเกิดเชื้อร้าและปัญหาสุขภาพของพืชค่าเฉลี่ยความชื้นสัมพัทธ์ 75.47% ถือว่าต่ำกว่าจุดอื่น ๆ บ่งบอกถึงสภาพแวดล้อมที่มีความชื้นพอสมควรแต่ไม่มากเกินไป ซึ่งอาจเหมาะสมสำหรับพืชที่ต้องการความชื้นระดับปานกลาง ไม่สูงมากเปรียบเทียบกับมาตรฐาน: 17.83 บ่งชี้ถึงความผันผวนของค่าความชื้นที่สูงที่สุดในทุกจุดที่วัดได้ แสดงให้เห็นถึงความไม่เสถียรของกระบวนการคุณภาพ ความชื้นที่มากกว่าจุดอื่น ซึ่งอาจส่งผลกระทบต่อพืชหรือสัตว์ที่ต้องการความชื้นคงที่

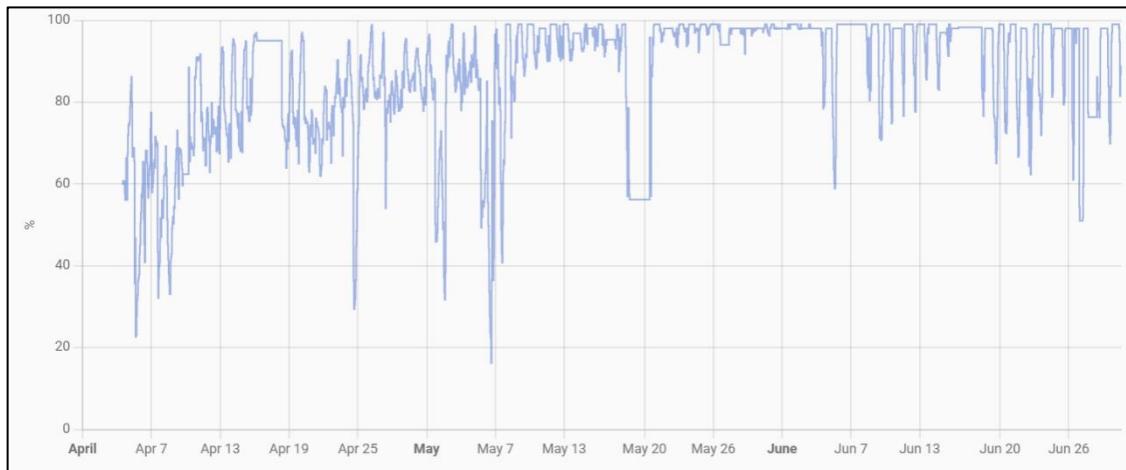


รูปที่ 4.18 ความชื้นสัมพัทธ์จุดที่ 4

4.3.2.5 ความชื้นสัมพัทธ์จุดที่ 5

ค่าความชื้นสัมพัทธ์ต่ำสุด: 16.22% แสดงถึงช่วงที่ความชื้นลดลงอย่างมาก ซึ่งอาจเกิดจากการระบายอากาศที่ดีเกินไปหรือสภาพอากาศภายนอกที่แห้งอย่างมาก การลดลงของความชื้นในระดับนี้อาจ

ไม่เหมาะสมสำหรับพืชที่ต้องการความชื้นสูง เพราะอาจทำให้พืชเกิดความเครียดและชะลอการเจริญเติบโต ค่าความชื้นสัมพัทธ์สูงสุด 99.00% บ่งบอกว่ามีช่วงเวลาที่ความชื้นใกล้ถึงจุดอิ่มตัว การที่ความชื้นสูงถึงระดับนี้อาจเกิดจากการควบคุมความชื้นที่ไม่เพียงพอหรือการปิดโรงเรือน ทำให้มีความเสี่ยงต่อการเกิดเชื้อราและปัญหาโรคพืช ค่าเฉลี่ยความชื้นสัมพัทธ์ 82.77% แสดงถึงสภาพแวดล้อมที่มีความชื้นสูงอย่างต่อเนื่อง หมายความว่าพืชที่ต้องการความชื้นสูง แต่ก็อาจเป็นสภาพที่ทำให้เชื้อราเติบโตได้่ายาก ไม่มีการจัดการที่ดีเบนนาตรฐาน 15.39 แสดงถึงความผันผวนของความชื้นในระดับค่อนข้างสูง แสดงถึงการที่ระบบควบคุมความชื้นยังไม่เสถียรเท่าที่ควร ทำให้ความชื้นเปลี่ยนแปลงไปอย่างมากในแต่ละช่วงเวลา

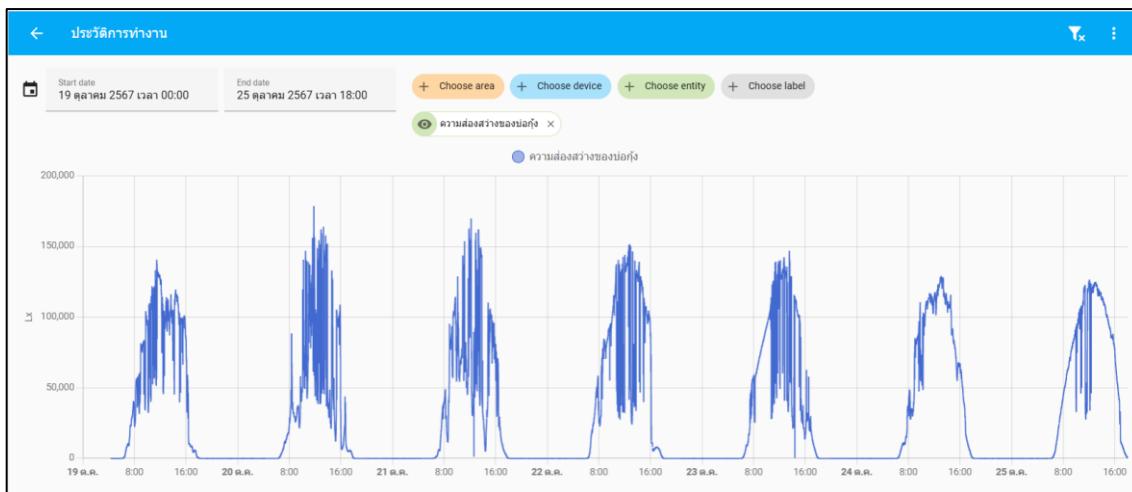


รูปที่ 4.19 ความชื้นสัมพัทธ์จุดที่ 5

4.4 ข้อมูลสภาพแวดล้อมภายในโรงเรือนบ่อ กุ้ง

4.4.1 ความส่องสว่างของบ่อ กุ้ง

กราฟนี้แสดงข้อมูลความส่องสว่างภายในโรงเรือนบ่อ กุ้ง ณ จุดตรวจวัดระหว่างวันที่ 19 - 25 ตุลาคม พ.ศ. 2567 โดยในแต่ละวันมีรูปแบบการเพิ่มขึ้นและลดลงของความส่องสว่างที่เป็นรอบเวลา ซึ่งบ่งบอกถึงการเปลี่ยนแปลงของแสงในช่วงเวลากลางวันและกลางคืน ช่วงเช้าความส่องสว่างเริ่มสูงขึ้นจนถึงช่วงกลางวัน และลดลงเมื่อเข้าใกล้ช่วงเย็นถึงค่ำต่ำสุดในตอนกลางคืน สอดคล้องกับการขึ้นและตกของดวงอาทิตย์ ขณะเดียวกัน ในแต่ละวันมีการเพิ่มขึ้นของความส่องสว่างในลักษณะที่ใกล้เคียงกัน ยกเว้นในบางวัน เช่น วันที่ 24 ตุลาคม ที่ค่าความส่องสว่างดูเหมือนจะมีการผันผวนมากขึ้นดังรูปที่ 4.20 ความส่องสว่างโรงเรือนบ่อ กุ้ง อาจเกิดจากสภาพอากาศที่มีเมฆมากหรือฝนตก โดยมีค่าความส่องสว่างสูงสุดอยู่ที่ 178,197 Lux ณ เวลา 11:49:23 น. ของวันที่ 20 ตุลาคม 2567



รูปที่ 4.20 ความส่องสว่างโรงเรือนบ่อ กุ้ง

4.4.2 Dissolved oxygen ของบ่อ กุ้ง

กราฟนี้แสดงข้อมูล DO ของน้ำภายในโรงเรือนบ่อ กุ้ง ณ จุดตรวจวัดระหว่างวันที่ 19 - 25 ตุลาคม พ.ศ. 2567 ส่วนใหญ่ค่าอยู่ระหว่าง 8.5 - 9.0 mg/L ซึ่งเป็นปกติ คุณภาพน้ำที่ดีและมีอوكซิเจนเพียงพอต่อสิ่งมีชีวิตน้ำ มีบางช่วงที่ DO ลดลงอย่างรวดเร็ว เช่น ในวันที่ 22 และ 23 ตุลาคม คาดว่าอาจเกิดจากการเพิ่มน้ำของสารอินทรีย์ ปัญหาการหมุนเรียนน้ำ หรืออุณหภูมิที่สูงขึ้น โดยรวมแนวโน้มค่า DO ค่อนข้างเสถียรดังรูปที่ 4.21 Dissolved oxygen โรงเรือนบ่อ กุ้ง โดยมีค่า DO สูงสุดอยู่ที่ 9.2 mg/L ณ เวลา 8:13:32 น. ของวันที่ 25 ตุลาคม 2567 ร่วมกับเวลา 10:43:12 น. ของวันที่ 25 ตุลาคม 2567

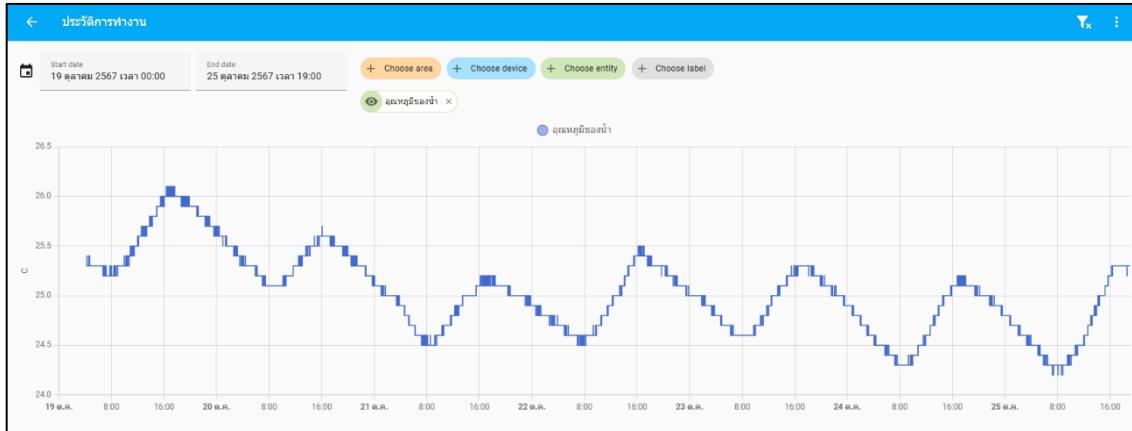


รูปที่ 4.21 Dissolved oxygen โรงเรือนบ่อ กุ้ง

4.4.3 อุณหภูมิน้ำของบ่อ กุ้ง

กราฟนี้แสดงข้อมูลอุณหภูมิของน้ำภายในบ่อ กุ้ง ณ จุดตรวจวัดระหว่างวันที่ 19 - 25 ตุลาคม พ.ศ. 2567 แสดงให้เห็นการเปลี่ยนแปลงตามวัฏจักรกลางวันและกลางคืน โดยอุณหภูมิสูงขึ้นในช่วงกลางวัน สูงสุดประมาณ 26 °C และลดลงในช่วงกลางคืนต่ำสุดประมาณ 24.5 °C ดังรูปที่ 4.22 อุณหภูมิน้ำของบ่อ

กุ้ง ซึ่งเป็นช่วงที่เหมาะสมสำหรับการเลี้ยงกุ้ง อุณหภูมิอยู่ในระดับที่เสถียรและไม่มีการเปลี่ยนแปลงรุนแรง เนื่องจากกุ้งส่วนใหญ่เติบโตได้ดีในน้ำที่อุณหภูมิประมาณ 24-30 °C

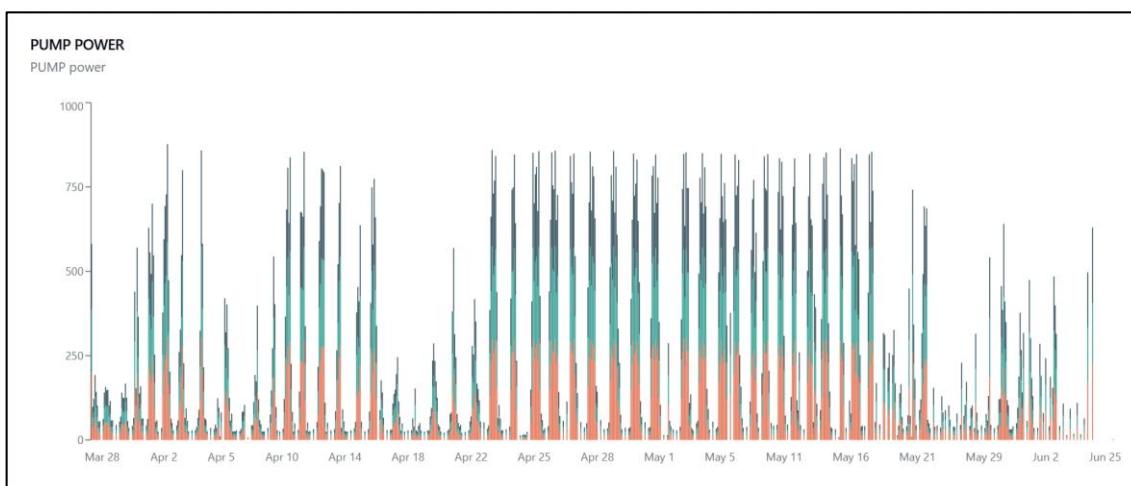


รูปที่ 4.22 อุณหภูมน้ำของปอกกุ้ง

4.5 การใช้พลังงาน

4.5.1 การใช้พลังงานของโรงเรือนกัญชา

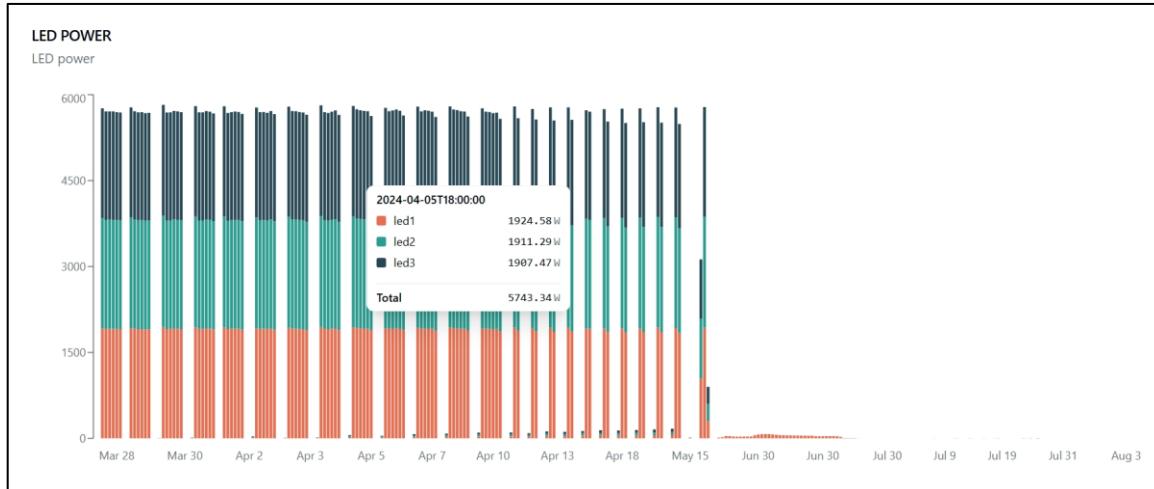
ผลการทดลองพบว่าในช่วง 3 เดือนของการเพาะปลูก โดยมีค่าเฉลี่ยกำลังไฟฟ้ามากที่สุด คือ 335.57 W ในเดือนที่สอง รองลงมาคือ 270.24 W, 157.24 W ในเดือนแรก และเดือนที่สาม ตามลำดับ ดังรูปที่ 4.23 กำลังไฟฟ้าของปั๊ม การใช้พลังงานไฟฟ้าภายใน 3 เดือนของการเพาะปลูกพบว่ามี การใช้พลังงานจากปั๊มคือ 87.28 kWh, 77.85 kWh และ 4.56 kWh ตามลำดับคิดเป็นพลังงานที่ใช้ ทั้งหมดของปั๊มภายใน 3 เดือนของการเพาะปลูกคือ 169.70 kWh พบว่าการใช้พลังงานมีปริมาณมากสุด ในช่วงเดือนแรกและลดลงหลังจากเดือนแรกอย่างมีนัยสำคัญ



รูปที่ 4.23 กำลังไฟฟ้าของปั๊ม

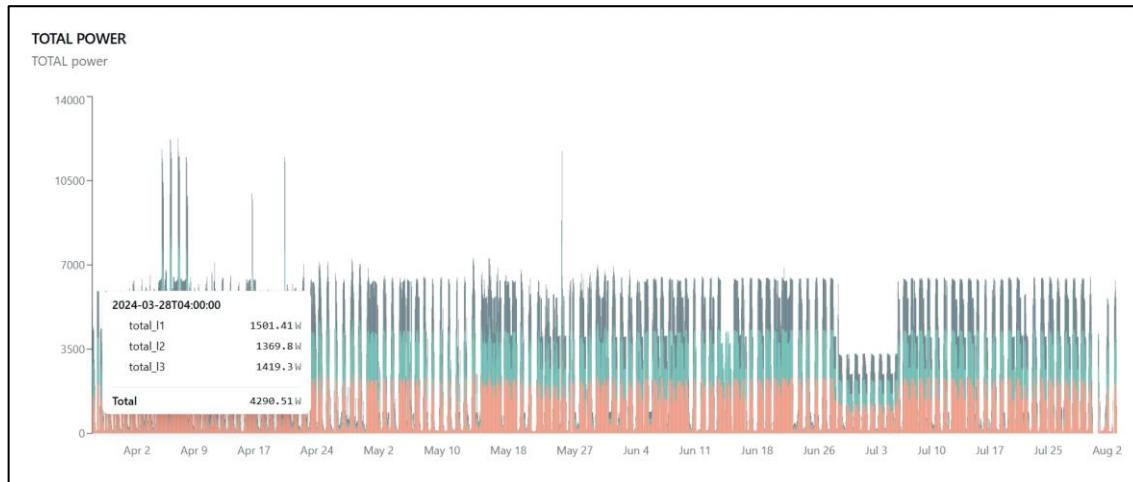
ผลการทดลองพบว่าในช่วง 3 เดือนของการเพาะปลูก พบร่วางในช่วง 3 เดือนของการปลูก มีการใช้ พลังงานจากระบบแสงสว่างคือ 423.16 kWh, 9.81 kWh, 1.48 kWh ตามลำดับดังรูปที่ 4.24 กำลังไฟฟ้า ของระบบไฟฟ้าแสงสว่าง โดยพบว่ามีการใช้พลังงานจากแสงสว่างมีสูงมากในช่วงเดือนแรกของการปลูก

และลดลงจนใกล้เคียงกับค่าศูนย์ ในช่วงเจริญเติบโต และช่วงเก็บเกี่ยวตามลำดับ โดยมีการใช้พลังงานจากระบบแสงสว่างทั้งหมด 434.46 kWh



รูปที่ 4.24 กำลังไฟฟ้าของระบบไฟฟ้าแสงสว่าง

ผลการทดลองพบว่าในช่วง 3 เดือนของการเพาะปลูก พบร่วมในช่วง 3 เดือนของการปลูก มีการใช้พลังงานทั้งระบบคือ 2,914.21 kWh, 2,779.34 kWh, 2,905.20 kWh ตามลำดับดังรูปที่ 4.25 พลังงานไฟฟ้าของโรงเรือน โดยมีการใช้พลังงานจากทั้งระบบทั้งหมด 8,598.75 kWh เหลือการใช้พลังงานต่อเดือนคือ 2,866.25 kWh โดยพบว่ามีการใช้พลังงานมีความผันผวนเล็กน้อยในระดับการใช้พลังงานในแต่ละเดือน แต่โดยรวมแล้วอยู่ในระดับที่ใกล้เคียงกัน โดยเดือนแรกและเดือนที่สามมีการใช้พลังงานใกล้เคียงกันและสูงกว่าตลอด 3 เดือน



รูปที่ 4.25 พลังงานไฟฟ้าของโรงเรือน

4.5.1.1 การคำนวณหากการใช้พลังงานจากพัดลมระบบบายอากาศ

จากการสำรวจพบว่า มีระบบที่ทำงานภายในโรงเรือนกัญชาทั้งหมด 3 ระบบหลักๆ คือ ระบบปั๊ม ระบบแสงสว่าง ระบบการระบายอากาศ (พัดลม) เมื่อเราทราบพลังงานที่ใช้ภายในโรงเรือนทั้งหมด พลังงานจากระบบปั๊ม ระบบแสงสว่างจะสามารถหาพลังงานส่วนที่เหลือได้เป็นพลังงานของระบบการระบายอากาศ (พัดลม) ได้ดังดังสมการที่ 4.1

$$\text{พลังงานระบบระบายอากาศ} = \text{พลังงานทั้งหมด}-\text{พลังงานงานส่วนอื่นๆ} \quad \text{สมการที่ 4.1}$$

$$\text{พลังงานระบบระบายอากาศ} = \text{พลังงานทั้งหมด}-(\text{พลังงานระบบปั๊ม}+\text{พลังงานระบบแสงสว่าง})$$

ตารางที่ 4.3 ข้อมูลการใช้พลังงานของโรงเรือนกัญชา

	เดือนที่ 1	เดือนที่ 2	เดือนที่ 3	รอบเพาะปลูก
พลังงานทั้งหมด (kWh)	2,914.21	2,779.34	2,905.20	8,598.75
พลังงานระบบปั๊ม (kWh)	87.28	77.85	4.56	169.70
พลังงานระบบแสงสว่าง (kWh)	423.16	9.81	1.48	434.46
พลังงานจากพัดลม (kWh)	2,403.77	2,691.68	2,899.16	7,994.61

พัดลมมีการใช้พลังงานตลอดสามเดือน โดยเดือนที่ใช้พลังงานมากที่สุดคือเดือนที่ 3 คือ 2,899.16 kWh และเดือนที่ใช้พลังงานน้อยที่สุดคือเดือนที่ 1 คือ 2,403.77 kWh พัดลมใช้พลังงานทั้งหมดคือ 7,994.61 kWh ดังตารางที่ 4.3 ข้อมูลการใช้พลังงานของโรงเรือนกัญชา การใช้พลังงานของพัดลมโดยเฉลี่ย 2,664.87 kWh ต่อเดือนการใช้พลังงานของพัดลมคิดเป็นส่วนใหญ่ของพลังงานรวมทั้งหมด

4.5.1.2 การใช้พลังงานต่อน้ำหนักผลผลิต

การเพาะปลูกภายในโรงเรือนที่มีการควบคุมสภาพแวดล้อม การใช้พลังงานในการผลิตกล้ายเป็นประเด็นสำคัญที่ต้องได้รับการวิเคราะห์และจัดการอย่างมีประสิทธิภาพ การปลูกกัญชาในโรงเรือนมักใช้พลังงานมาก โดยเฉพาะในด้านระบบไฟฟ้า ระบบควบคุมอุณหภูมิ และการจัดการสภาพแวดล้อมอื่น ๆ ที่เกี่ยวข้อง การคำนวณพลังงานที่ใช้ต่อหน่วยผลผลิต (kWh/g) จึงเป็นตัวชี้วัดสำคัญในการประเมินประสิทธิภาพการผลิต

ตารางที่ 4.4 ผลผลิตตลอด 3 เดือน

น้ำหนักผลผลิตสด (g)	
Cb3g	617.05
CcBD	268.14
FH	348.26
รวม	1233.45

น้ำหนักผลผลิตของกัญชาจากสามสายพันธุ์ในช่วง 3 เดือน ได้แก่ Cb3g, CcBD, และ FH พร้อมกับการคำนวณพลังงานที่ใช้ในการผลิตกัญชาต่อกรัม ผลผลิตของกัญชาจากสามสายพันธุ์คือ Cb3g มีน้ำหนักผลผลิตทั้งหมด 617.05 g ซึ่งถือว่าเป็นผลผลิตที่มากที่สุดในบรรดาทั้งสามสายพันธุ์ CcBD มีน้ำหนักผลผลิต 268.14 g ซึ่งเป็นผลผลิตที่น้อยที่สุดเมื่อเทียบกับสายพันธุ์อื่น ๆ FH มีน้ำหนักผลผลิต 348.26 g อยู่ในระดับกลางเมื่อเทียบกับอีกสองสายพันธุ์ เมื่อรวมผลผลิตจากทั้งสามสายพันธุ์ น้ำหนักผลผลิตรวมทั้งหมดในช่วง 3 เดือนคือ 1,233.45 g หรือประมาณ 1.23 กิโลกรัม ซึ่งเป็นน้ำหนักผลผลิตรวมที่ได้จากการปลูกกัญชาในระยะเวลาหนึ่ง การคำนวณพลังงานต่อผลผลิต (kWh/g) เป็นการหาตัวชี้วัดที่สำคัญในการวัดประสิทธิภาพการใช้พลังงานในกระบวนการผลิต โดยพลังงานทั้งหมดที่ใช้ การผลิตกัญชาทั้งหมดในช่วง 3 เดือนนี้ใช้พลังงานไฟฟ้ารวมทั้งสิ้น 8,598.75 kWh น้ำหนักผลผลิตทั้งหมดจากทั้งสามสายพันธุ์รวมกันคือ 1,233.45 g เมื่อเราคำนวณพลังงานทั้งหมดมาแบ่งด้วยน้ำหนักผลผลิต จะได้ค่าเฉลี่ยพลังงานที่ใช้ต่อการผลิตกัญชาหนึ่ง g

$$\text{พลังงาน/ผลผลิต} = \frac{\text{พลังงานทั้งหมด (kWh)}}{\text{น้ำหนักผลผลิตสด (g)}}$$

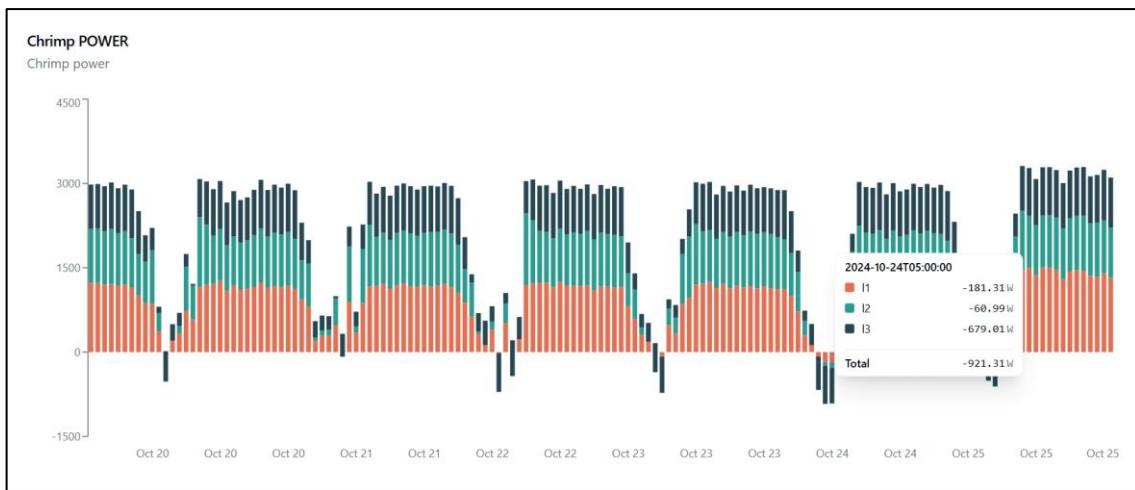
$$\text{พลังงาน/ผลผลิต} = \frac{8598.75 \text{ kWh}}{1233.45 \text{ g}}$$

$$\text{พลังงาน/ผลผลิต} = 6.97 \text{ kWh/g}$$

ผลการคำนวณแสดงให้เห็นว่า พลังงานที่ใช้ในการผลิตกัญชาอยู่ที่ 6.97 kWh/g นั่นหมายความว่าในทุก ๆ การผลิตกัญชา 1 g ต้องใช้พลังงานไฟฟ้าประมาณ 6.97 kWh

4.5.2 การใช้พลังงานของโรงเรือนบ่อกุ้ง

การใช้พลังงานในบ่อกุ้ง พบว่าการใช้พลังงานรวมในช่วงหนึ่งสัปดาห์อยู่ที่ 328.43 kWh โดยสังเกตเห็นการไหลาย้อนกลับของกระแสไฟฟ้าบางช่วง ซึ่งเกิดจากพลังงานที่ผลิตจากโซลาร์เซลล์ ทำให้มีการส่งพลังงานกลับเข้าสู่ระบบในบางช่วงเวลาที่พลังงานผลิตได้มากเกินความต้องการ



รูปที่ 4.26 การใช้พลังงานรวมของป่ากุ้ง

4.6 ประสิทธิภาพของระบบ

ในการทดสอบประสิทธิภาพ ได้ทำการสร้าง ผู้ใช้งานจำลอง Virtual Users (VUs) จำนวน 100, 500, 1000, 3000 และ 5000 คนตามลำดับ เพื่อหาความเร็วในการตอบสนอง อัตราส่วนการขอข้อมูลสำเร็จ และ เปอร์เซนต์การขอข้อมูลสำเร็จ พบร้า สามารถรองรับผู้ใช้งานพร้อมกันอยู่ที่ 3000 คน รองรับคำขอแสดงข้อมูลได้ 1000 คำขอต่อวินาที ตอบสนองได้รวดเร็วอยู่ที่ 1.08 วินาที

ตารางที่ 4.5 ผลการทดสอบประสิทธิภาพ

VUs	Req/s	Avg. response time(s)	Success / req	%Success rate
100	90.73	1.08	2800/2800	100%
500	439.38	1.01	13647/13647	100%
1000	823.81	1.18	25680/25680	100%
3000	1122.47	2.56	35944/35944	100%
5000	781.77	5.62	27641/28618	98.13%

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

จากการนำข้อมูลที่ได้จากการตรวจวัดพลังงานที่ใช้ภายในโรงเรือน ผ่าน ESP32 มาจัดเรียงข้อมูล และ จัดข้อมูลได้มาตลอดระยะเวลาโครงการ 3 เดือน เป็นข้อมูลแบ่งออกเป็น 2 ส่วนใหญ่ๆซึ่งได้แก่ การใช้พลังงาน ภายในระบบที่จำแนกออกเป็นแต่ละส่วนของการใช้พลังงานทั้งหมด 8,598.76 kWh โดยแบ่ง 614.06 kWh ได้ ออกมาเป็น ระบบแสงไฟฟ้าส่องสว่าง มีการใช้พลังงาน 72.3% และ ระบบปั๊มน้ำ มีการใช้พลังงาน 27.7% รวมทั้ง สองระบบคิดเป็น 7.1% ของพลังงานที่ใช้ทั้งหมดภายในโรงเรือนกัญชา การใช้พลังงานภายในโรงเรือนโดยจำแนก ตามเวลา พบร่วมกันว่าการใช้แสงสว่างถึงจุดสูงสุดในเดือนแรกที่ช่วงการเป็นต้น อ่อนชี้ให้เห็นความต้องการแสงของกัญชาจะ ส่งผลต่อการเจริญเติบโตอย่างมาก และค่อยๆ ลดลงหลังจากนั้น ในทำนองเดียวกันการใช้พลังงานจากปั๊มจะสูง ในช่วงสองเดือนแรกของการปลูก แต่ลดลงอย่างเห็นได้ชัดในเดือนที่สาม รูปแบบนี้เน้นให้เห็นถึงความต้องการ พลังงานที่แตกต่างกันของเรื่องผลกระทบต่อการเจริญเติบโต การใช้พลังงานเฉลี่ยในการผลิตกัญชาในโรงเรือนตลอดระยะเวลา 3 เดือนอยู่ที่ 6.97 kWh ต่อตัวของผลผลิต ซึ่งแสดงให้เห็นว่ากระบวนการผลิตกัญชาต้องใช้พลังงานค่อนข้างสูง ในส่วนของโรงเรือนป่ากุ้ง ค่าความส่องสว่างสอดคล้องกับการขึ้นและตกของดวงอาทิตย์ ในแต่ละวันมีการเพิ่มขึ้นของความส่องสว่างในลักษณะที่ใกล้เคียงกัน ยกเว้นในวันที่ 24 ตุลาคม ที่ค่าความส่องสว่างดูเหมือนจะมีการผันผวนมากขึ้น DO ของน้ำส่วนใหญ่คงที่อยู่ระหว่าง 8.5 - 9.0 mg/L ซึ่งบ่งบอกถึงคุณภาพน้ำที่ดีและมีออกซิเจนเพียงพอต่อสิ่งมีชีวิตน้ำ มีบางช่วงที่ DO ลดลงอย่างรวดเร็ว เช่น ในวันที่ 22 และ 23 ตุลาคม คาดว่าอาจเกิดจากการเพิ่มขึ้นของสารอินทรีย์, ปัญหาการหมุนเวียนน้ำ หรืออุณหภูมิที่สูงขึ้น โดยรวมแนวโน้มค่า DO ค่อนข้างเสถียรเหมาะสมต่อการเลี้ยงกุ้ง ส่วนของอุณหภูมน้ำ เปลี่ยนแปลงตามวัฏจักร กลางวัน-กลางคืน โดยอุณหภูมิสูงขึ้นในช่วงกลางวัน สูงสุดประมาณ 26 °C และลดลงในช่วงกลางคืน ต่ำสุดประมาณ 24.5 °C เป็นช่วงที่เหมาะสมสำหรับการเลี้ยงกุ้ง การใช้พลังงาน กุ้ง พบร่วมกันว่าการใช้พลังงานรวมในช่วงหนึ่งสัปดาห์อยู่ที่ 328.43 kWh โดยสังเกตเห็นการไหลย้อนกลับของกระแสไฟฟ้าบางช่วง ซึ่งเกิดจากพลังงานที่ผลิตจากโซลาร์เซลล์ ทำให้มีการส่งพลังงานกลับเข้าสู่ระบบในบางช่วงเวลาที่พลังงานผลิตได้มากเกินความต้องการ

5.2 ข้อเสนอแนะ

การตรวจวัดพลังงานในโรงเรือนเป็นกระบวนการที่สำคัญต่อการจัดการทรัพยากรอย่างมีประสิทธิภาพ ข้อมูลที่ได้จากการตรวจวัดสามารถช่วยให้ทราบถึงการใช้พลังงานในระบบต่าง ๆ และช่วยวางแผนในการปรับปรุงประสิทธิภาพการผลิตได้อย่างแม่นยำ เพื่อเพิ่มความสามารถในการตรวจวัดและการควบคุม เราสามารถพัฒนาและเพิ่มจุดตรวจวัดพลังงานในระบบอื่น ๆ เพื่อให้ครอบคลุมมากขึ้นเพื่อนำข้อมูลมาปรับปรุงประสิทธิภาพการใช้พลังงานในอนาคต เช่น การพัฒนาระบบควบคุมการใช้พลังงานในแต่ละช่วงการเจริญเติบโตให้เหมาะสม หรือนำพลังงานทดแทนมาใช้ในโรงเรือนเพื่อเพิ่มความยั่งยืนในการผลิต จากข้อมูลของโรงเรือนบ่อ กุ้ง ควรเฝ้าระวังช่วงที่ DO ลดลงอย่างกะทันหันเพื่อป้องกันปัญหาที่อาจเกิดขึ้นและตรวจสอบสภาพแวดล้อมของแหล่งน้ำอย่างต่อเนื่อง ส่วนของอุณหภูมิของบ่อ กุ้งควรเฝ้าระวังช่วงกลางคืนหากอุณหภูมิลดลงต่ำกว่า 24 °C ต่อเนื่อง อาจต้องมีมาตรการควบคุม เช่น การใช้เครื่องทำความร้อน เพื่อป้องกันความเสี่ยงต่อกุ้ง ติดตามการเปลี่ยนแปลงตามฤดูกาลเตรียมพร้อมสำหรับการเปลี่ยนแปลงตามฤดูกาล เพราะอุณหภูมิลดลงมากขึ้นในช่วงฤดูหนาว และมีกระแสไฟฟ้าเหลืออยู่กลับเข้าไปในระบบของโรงเรือนบ่อ กุ้งเนื่องจากมีการติดตั้งระบบผลิตพลังงานไฟฟ้าจากแสงอาทิตย์มีพลังงานในบางช่วงมากกว่าที่ต้องการใช้ ทำให้ การติดตั้งแบตเตอรี่สำหรับพลังงานส่วนเกิน อาจทำให้ระบบใช้พลังงานได้คุ้มค่ามากกว่าเดิม และเพื่อประหยัดค่าใช้จ่ายและเพิ่มความเสถียรของระบบสำหรับプロジェกขนาดใหญ่ การใช้บริการคลาวด์ ด้วยการใช้ Docker container เพื่อทำการเพิ่มเสถียรภาพ เชิฟเวอร์ไม่ตับบอย การใช้ Docker อาจจะมีความยุ่งยากมากขึ้นถ้าเทียบกับการใช้ Raspberry Pi Homeassistant OS แต่ในเชิงการประหยัดต้นทุนและความเสี่ยรของระบบ การใช้ Docker มีความได้เปรียบอย่างเห็นได้ชัด

เอกสารอ้างอิง

- กระทรวงพลังงาน. (2015) แผนอนุรักษ์พลังงาน พ.ศ.2558–2579. แผนบูรณาการพลังงานระยะยาว.
www.eppo.go.th
- จิรพันธ์ พิมพล, วีระพล แก้วกำ, บรรดาล สังข์ครบ, อรพิน ชาญนำสิน, และวรพงศ์ ตั้งศรีรัตน์. (2023). ระบบตรวจสอบและควบคุมโรงเรือนปลูกผักแบบปิด. การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้า ครั้งที่ 15, 662-665
- บุญญฤทธิ์ วงศ์วัน, สมบัติย์ มงคลชัยชนะ, เอกรัฐ ชะอุ่มເວີດ, ศุภชัย ชุมนุมวัฒน์, ເດືອນແຮມ ແພ່ງເກີຍ. (2016). ประยุกต์ใช้พลังงานไฟฟ้าจากแสงอาทิตย์ในโรงเรือนไฮโดรโปนิกส์ของกลุ่มวิสาหกิจชุมชนผักปลอดภัย. การประชุมวิชาการวิจัยและนวัตกรรมสร้างสรรค์ ครั้งที่ 5, 1,1070-1077
- olson กองมนี, พัชรี ยางยืน, เสกสรรร พันทึก. (2020). ระบบโรงเรือนอัจฉริยะ เพื่อการผลิตผักอินทรีย์. สำนักหอสมุด มหาวิทยาลัยแม่โจ้, 635-483
- อุษาวดี ตันติราษฎร์. (2016). การศึกษาอัตราการระบายน้ำอากาศธรรมชาติ ของโรงเรือนเกษตรกรรม. วารสารวิทยาศาสตร์ มศว. ปีที่ 22, 1, หน้า 5-13
- Kannan, M., Elavarasan, G., & Balamurugan, A. Dhanusiya. B, & Freedon D. (2022). Hydroponic farming A state of art for the future agriculture, *International Conference*, 63,118-166.
- Kenny, K., Tan, P., & Maybelline, G., (2023). SMART GROW – Low-cost automated hydroponic system for urban farming, *Universiti Malaysia Sarawak*,
- Samira B., Dariush, H., & Ezatollah, K. (2022). Determining and validating criteria to measure energy consumption sustainability in agricultural greenhouses, 185, 122-077.
- Ilham, I., Rachid, T., Nora, A., & Hind, K. (2023). Design of a low-cost active and sustainable autonomous system for heating agricultural greenhouses, *Faculty of Sciences* 9, 014-582.
- Parastoo, M., Ramin, R. (2023). Optimal design and operation of solar energy system with heat storage for agricultural greenhouse heating, *Sharif University*. 18, 100-353.

ภาคผนวก ก
ข้อมูลผลการวิจัย



ภาคผนวก ก. ข้อมูลการใช้พลังงาน

ภาคผนวก ข
ประวัติผู้จัดทำโครงการ

ประวัติผู้จัดทำโครงการคนที่ 1

ชื่อ-สกุล นายกรดร น้ำสงค์
รหัสนักศึกษา 6415123324



ประวัติการศึกษา

ระดับการศึกษา	โรงเรียน	ปีการศึกษาที่สำเร็จการศึกษา
มัธยมศึกษาปีที่ 6	เมืองพัทยา 11	2564

สถานที่ติดต่อได้สะดวก

ที่อยู่	17/100 หมู่ที่ 5 ตำบลหนองปรือ อำเภอบางละมุง จังหวัดชลบุรี
รหัสไปรษณีย์	20150
เบอร์โทรศัพท์	099 620 8471
Email address	mju6415123324@mju.ac.th
Facebook	Paradon Namsong

ประวัติผู้จัดทำโครงการคนที่ 2

ชื่อ-สกุล นายภราดร ลุงช้อย
 รหัสนักศึกษา 6415123325



ประวัติการศึกษา

ระดับการศึกษา	โรงเรียน	ปีการศึกษาที่สำเร็จการศึกษา
มัธยมศึกษาปีที่ 6	ฝางชนูปถัมภ์	2564

สถานที่ติดต่อได้สะดวก

ที่อยู่	78/10 หมู่ 2 ตำบลท่าศาลา อำเภอเมืองเชียงใหม่ จังหวัดเชียงใหม่
รหัสไปรษณีย์	50000
เบอร์โทรศัพท์	095 447 8641
Email address	paradorn3237@gmail.com
Facebook	Dxsebxy Swc

ประวัติผู้จัดทำโครงการคนที่ 3

ชื่อ-สกุล วัชรพล คำลัน
รหัสนักศึกษา 6415123335



ประวัติการศึกษา

ระดับการศึกษา	โรงเรียน	ปีการศึกษาที่สำเร็จการศึกษา
มัธยมศึกษาปีที่ 6	สารสิทธิ์พิทยาลัย	2564

สถานที่ติดต่อได้สะดวก

- ที่อยู่ 46 หมู่.7 ตำบลท่าวุ้ง อำเภอท่าวุ้ง จังหวัดพบูรี
รหัสไปรษณีย์ 15150
เบอร์โทรศัพท์ 061 228 1100
Email address tas1352546@gmail.com
Facebook Watcharapon Khanlon