



การใช้เทคโนโลยีไอโอทีควบคุมโรงเพาะเห็ดนางฟ้าภูฏานระบบปิดผ่านสมาร์ทโฟนเพื่อป้องกัน
อันตรายต่อผู้เพาะเห็ด

Internet of Things Technology Control in Smart Farms Bhutan Oyster Mushroom
Close System through smartphones to prevent harm to mushroom growers

รุจิมาส ปิยมงคลชัย

ศุภิสรา เกิดแจ่ม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

ปีการศึกษา 2565

การใช้เทคโนโลยีไอโอทีควบคุมโรงเพาะเห็ดนางฟ้าภูฏานระบบปิดผ่านสมาร์ทโฟนเพื่อป้องกัน
อันตรายต่อผู้เพาะเห็ด

รุจิมาส ปิยมงคลชัย
ศุภิสรา เกิดแจ่ม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
ปีการศึกษา 2565

INTERNET OF THINGS TECHNOLOGY CONTROL IN SMART FARMS BHUTAN
OYSTER MUSHROOM CLOSE SYSTEM THROUGH SMARTPHONES TO PREVENT
HARM TO MUSHROOM GROWERS

MISS RUJIMAS PIYAMONGKOLCHAI
MISS SUPHISARA KOETCHEAM

THIS PROJECT SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIRMENTS FOR
THE BACHELOR DEGREE OF SCIENCE
DEPARTMENT OF TECHNOLOGY COMPUTER
FACULTY OF SCIENCE
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI
YEAR 2022

หัวข้อปริญญานิพนธ์	การใช้เทคโนโลยีไอโอทีที่ควบคุมโรงเพาะเห็ดนางฟ้าภูฐานระบบปิดผ่านสมาร์ตโฟนเพื่อป้องกันอันตรายต่อผู้เพาะเห็ด
นักศึกษา	นางสาวรุจิมาศ ปิยมงคลชัย รหัส 116210906075-3 นางสาวศุภิสรา เกิดแจ่ม รหัส 116210906076-1
อาจารย์ที่ปรึกษา	อาจารย์กิริติบุตร กาญจนเสถียร
ปีการศึกษา	2565

บทคัดย่อ

โครงการวิจัยระดับปริญญาตรีเรื่อง โรงเพาะเห็ดนางฟ้าภูฐานระบบปิดผ่านเทคโนโลยีไอโอที สาขาวิชาเทคโนโลยีคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี จัดทำโดยมีวัตถุประสงค์เพื่อนำเทคโนโลยีไอโอทีมาประยุกต์ใช้กับการทำเกษตรกรรมให้เกิดประโยชน์และเป็นเครื่องมือที่ช่วยอำนวยความสะดวกให้กับเกษตรกรที่ทำการเพาะเห็ดนางฟ้าภูฐานขายและผู้ที่ยากทำอาชีพเพาะเห็ดนางฟ้าภูฐานขายเป็นรายได้เสริมสามารถนำเอาเทคโนโลยีนี้ไปประยุกต์ใช้ได้โดยระบบที่จัดทำขึ้นนั้นจะมีระบบที่ช่วยในการดูแลการปลูกเห็ดนางฟ้าภูฐาน เช่น ระบบจะสามารถเพิ่มความชื้นโดยการรดน้ำได้โดยอัตโนมัติ ระบบจะสามารถระบายอากาศได้แบบอัตโนมัติเมื่ออุณหภูมิภายในโรงเรือนสูงเกินค่ามาตรฐานในการออกดอกของเห็ดนางฟ้าภูฐาน ระบบจะสามารถบอกค่าก๊าซภายในโรงเรือนเพาะเห็ดนางฟ้าภูฐานที่เป็นอันตรายต่อมนุษย์ได้

กิตติกรรมประกาศ

โครงการวิจัยระบบปัญญาตรี เรื่อง โรงเรือนเพาะเห็ดนางฟ้าภูฏานระบบปิดควบคุมผ่านเทคโนโลยีไอโอที Internet of Things Technology Control in Smart Farms Bhutan Oyster Mushroom Close System through smartphones to prevent harm to mushroom growers ณ โอกาสนี้ ผู้วิจัยขอขอบพระคุณ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี ซึ่งอำนวยความสะดวกในเรื่องของสถานที่ที่ใช้สำหรับการดำเนินโครงการ และขอกราบขอบพระคุณ อาจารย์กิริติบุตร กาญจนเสถียร ที่ได้สละเวลาให้ ข้อเสนอแนะแนวทางอันเป็นประโยชน์ ตลอดจนช่วยแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่และให้ ความห่วงใยเสมอมา

ขอขอบพระคุณอาจารย์และเจ้าหน้าที่ที่สาขาวิชา เทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรีทุกท่านที่ได้ประสิทธิประสาทวิชาให้นักศึกษาเป็นผู้ที่มีความรู้ซึ่งทำให้การจัดปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

ท้ายที่สุดนี้คุณประโยชน์อันพึงได้จากปริญญานิพนธ์นี้คณะผู้จัดทำขอมอบให้แด่บิดา มารดา และอาจารย์ทุกท่านที่ได้ประสิทธิประสาทวิชาให้แก่นักศึกษาทุกคน

คณะผู้จัดทำ

รุจิมาส ปิยมงคลชัย

ศุภิสรา เกิดแจ่ม

สารบัญ

	หน้า
บทคัดย่อ	ง
กิตติกรรมประกาศ	จ
สารบัญ	ฉ
สารบัญตาราง	ช
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของการวิจัย	2
1.4 เครื่องมือที่ใช้ในงานวิจัย	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ทฤษฎีที่เกี่ยวข้อง	4
2.2 งานวิจัยที่เกี่ยวข้อง	7
บทที่ 3 วิธีดำเนินงานวิจัย	9
3.1 ศึกษาและเก็บรวบรวมข้อมูล	9
3.2 MOCK UP ของแอปพลิเคชัน	10
3.3 การพัฒนาในส่วนของฮาร์ดแวร์	17
3.4 อุปกรณ์ที่ใช้ในการจัดทำโครงการ	21
3.5 ทดสอบระบบอุปกรณ์ฮาร์ดแวร์	24
3.6 การทดสอบในส่วนของซอฟต์แวร์	26
3.7 การเก็บฐานข้อมูล	32
บทที่ 4 ผลการทดลอง	34
4.1 ผลการทดลองการเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐานระยะเวลา 3 วัน	34
4.2 ผลการทดลองการเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐานระยะเวลา 5 วัน	35
4.3 ผลการทดลองการเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐาน 7 วัน หลังเพาะ	35

4.4 ผลการเกิดโรคเชื้อราเขียว	36
4.5 ผลการเสียหายของก้อนเห็ดนางฟ้าภูฐานที่เกิดจากแมลง	36
4.6 คุณภาพของเห็ดนางฟ้าภูฐาน	37
บทที่ 5 สรุปและข้อเสนอแนะ	38
5.1 สรุปและวิจารณ์ผลการทดลอง	38
5.2 ข้อเสนอแนะและแนวทางในการพัฒนา	40
บรรณานุกรม	41
ภาคผนวก ก	43
ภาคผนวก ข	52
ภาคผนวก ค	83
ภาคผนวก ง	88
ประวัติผู้จัดทำปริญญานิพนธ์	92

สารบัญตาราง

	หน้า
ตารางที่ 3-1 Gantt chart แสดงระยะเวลาในการดำเนินการ	9
ตารางที่ 3-2 แบบทดสอบประสิทธิภาพการทำงานของระบบโรงเพาะเห็ดนางฟ้าภูฐาน	24
ตารางที่ 3-3 ตารางการทดสอบในส่วนของซอฟต์แวร์	26
ตารางที่ 4-1 ผลการทดลองการเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐาน 3 วัน	34
ตารางที่ 4-2 ผลการทดลองการเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐาน ระยะเวลา 5 วัน	35
ตารางที่ 4-3 ผลการทดลองการเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐาน ระยะเวลา 7 วัน	35
ตารางที่ 4-4 บันทึกผลการเกิดโรคเชื้อราเขียว	36
ตารางที่ 4-5 บันทึกผลการเสียหายของก้อนเห็ดนางฟ้าภูฐานที่เกิดจากแมลง	36
ตารางที่ 4-6 บันทึกผลคุณภาพของเห็ดนางฟ้าภูฐาน	37

สารบัญรูป

หน้า

ภาพที่ 3-1 หน้าแรกของแอป เพื่อให้เลือกว่าจะ Sign Up (ลงทะเบียน) หรือ Log in (เข้าสู่ระบบ)	10
ภาพที่ 3-2 หน้า SIGN UP(ลงทะเบียน) สำหรับคนที่ยังไม่เคยเข้าใช้ระบบ	11
ภาพที่ 3-3 Log in (เข้าสู่ระบบ) เพื่อเข้าใช้งานแอปควบคุมโรงเพาะเห็ด	11
ภาพที่ 3-4 เมื่อเข้าสู่ระบบจะพบหน้าควบคุมค่าต่างๆภายในโรงเรือนดังนี้	12
ภาพที่ 3-5 Flowchart แสดงขั้นตอนการทำงานของระบบโรงเพาะเห็ดนางฟ้าภูฐาน	13
ภาพที่ 3-6 Context Diagram แสดงการทำงานของระบบโรงเพาะเห็ดนางฟ้าภูฐาน	14
ภาพที่ 3-7 Data Flow Diagram level1 แสดงการทำงานของระบบโรงเพาะเห็ดนางฟ้าภูฐาน	15
ภาพที่ 3-8 DATA FLOW DIAGRAM LEVEL2 แสดงการทำงานของระบบแอปพลิเคชันของโรงเพาะเห็ดนางฟ้าภูฐาน	15
ภาพที่ 3-9 ER-Diagram Model แสดงแบบจำลองโครงสร้างของแอปพลิเคชัน	16
ภาพที่ 3-10 โครงของโรงเรือนสำหรับเพาะเห็ด	17
ภาพที่ 3-11 ตำแหน่งของการติดตั้งเซนเซอร์ภายในโรงเรือน	18
ภาพที่ 3-12 โครงของโรงเรือนเมื่อคลุมสแลนแล้ว	18
ภาพที่ 3-13 โรงเรือนเมื่อคลุมพลาสติก	19
ภาพที่ 3-14 ตำแหน่งของการติดตั้งพัดลมระบายอากาศและปั้มน้ำ	19
ภาพที่ 3-15 ตำแหน่งของการติดตั้งหัวพ่นน้ำ	20
ภาพที่ 3-16 ตำแหน่งของการติดตั้งตู้คอนโทรล	20
ภาพที่ 3-17 Air quality sensor MQ 135 เซนเซอร์ตรวจวัดคุณภาพอากาศ	21
ภาพที่ 3-18 เซนเซอร์ DHT 22 แสดงการใช้วัดค่าความชื้นและอุณหภูมิ	21
ภาพที่ 3-19 พัดลมระบายอากาศขนาด 8 นิ้ว	22
ภาพที่ 3-20 มอเตอร์ปั้มน้ำ 10.8บาร์	22
ภาพที่ 3-21 หัวพ่นหมอกทองเหลือง	23
ภาพที่ 3-22 ทดสอบการทำงานของระบบหัวฉีดพ่นหมอก	25
ภาพที่ 3-23 หน้าแรกที่เปิดขึ้นมาเพื่อเลือก log in หรือ sign up	27
ภาพที่ 3-24 หน้าลงทะเบียน สำหรับผู้ที่ยังไม่มีบัญชี	28

ภาพที่ 3-25 หน้าเข้าสู่ระบบ สำหรับผู้ที่มีบัญชีผู้ใช้แล้ว	29
ภาพที่ 3-26 หน้าแสดงค่าต่างๆ ของระบบโรงเพาะเห็ด	30
ภาพที่ 3-27 ระบบมีการแจ้งเตือนเมื่อค่าก๊าซเกิน	31
ภาพที่ 3-28 การเก็บ log ต่างๆของเซนเซอร์จะถูกเก็บอยู่ใน Realtime database บน firebase	32
ภาพที่ 3-29 การเก็บรูปภาพที่จะแสดงบนแอปพลิเคชันถูกเก็บ อยู่ใน Storage บน firebase	32
ภาพที่ 3-30 การเก็บข้อมูลของการเข้าสู่ระบบถูกเก็บอยู่ใน Authentication บน firebase	32
ภาพที่ 3-31 การยิงค่าขึ้น Realtime datanase บน Firebase	33
ภาพที่ 4-1 การเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐาน 3 วัน หลังเพาะ	34
ภาพที่ 4-2 การเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐาน 5 วัน หลังเพาะ	35
ภาพที่ 4-3 การเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐาน 7 วัน หลังเพาะ	36

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันประเทศไทยมีจำนวนประชากร 9,267,103 คน ที่ทำอาชีพเกษตรกร มีเกษตรกรจำนวนมากเลือกทำการเกษตรแบบผสมผสาน (กระทรวงเกษตรและสหกรณ์, 2564) พร้อมทั้งมีอีกหลายบุคคลเลือกทำอาชีพเกษตรกรเป็นการหารายได้เสริม เนื่องจากเป็นอาชีพที่กำหนดต้นทุนได้ เป็นเกษตรกรรมที่ไม่ต้องพึ่งฟ้าพึ่งฝน ใช้น้ำที่ไม่เปลืองไม่จำเป็นต้องมีพื้นที่เยอะก็สามารถทำได้ ไม่ต้องตากแดด แต่อาชีพเพาะเห็ดก็ทำได้ไม่ง่าย การดูแลเพาะเห็ดนั้นเป็นเรื่องที่วุ่นวายเพราะต้องมีการดูแลทั้งอากาศและการรดน้ำเพื่อรักษาความชื้น ซึ่งบางครั้งต้องใช้เวลาเยอะ ซึ่งจะทำให้เกษตรกรที่ทำการเพาะเห็ดเสียเวลาในการดูแลส่วนอื่นในการทำเกษตร (ฟาร์มเห็ดกลางกรุง, 2564)

ในการเพาะเห็ดนั้นจะมีระยะในการเติบโตแบ่งออกเป็นสองระยะ คือ 1.ระยะการบ่มเชื้อเห็ด 2.ระยะเกิดดอก ซึ่งในระยะของการบ่มเชื้อเห็ดนั้นเกษตรกรที่ใช้การอัดก๊าซคาร์บอนไดออกไซด์เข้าไปในโรงเพาะเห็ดเพื่อให้เชื้อเห็ดเกิด ซึ่งเคยมีกรณีที่เกษตรกรได้ทำการอัดก๊าซคาร์บอนไดออกไซด์เข้าไปในโรงเพาะเห็ดและทำการเข้าไปตรวจสอบเห็ดด้วยตัวเองจนขาดอากาศหายใจจนถึงแก่ชีวิต 3 ราย (สำนักข่าวไทยรัฐ, 2564)

เห็ดนางฟ้าเป็นเห็ดที่นิยมรับประทานกันทั่วไป โดยเป็นเห็ดที่เกิดขึ้นเองตามธรรมชาติโดยจะเกิดเฉพาะ ช่วงฤดูกาลเท่านั้นเมื่อมีผู้นิยมบริโภคกันมากขึ้น จึงทำให้เกิดการพัฒนาไปสู่การเพาะเห็ดในเชิงการค้ามากขึ้น ([puechkaset](#), 2559)

แนวคิด Smart Farm หรือเกษตรอัจฉริยะ คือการทำการเกษตรที่นำเอาระบบเทคโนโลยีเข้ามาช่วยในการบริหารจัดการดูแลการเพาะปลูก รวมไปถึงกระบวนการผลิตเพื่อนำไปสู่การเกษตรเชิงธุรกิจ ด้วยการผสมผสานศาสตร์ทางวิทยาศาสตร์การเกษตร กับศาสตร์ทางวิศวกรรมเทคโนโลยีอิเล็กทรอนิกส์ และนำเอาเทคโนโลยีสมัยใหม่เข้ามาผสมผสานเข้ากับงานด้านการเกษตรเพื่อช่วยแก้ไขปัญหาต่างๆ ให้กับเกษตรกร

เทคโนโลยีฟาร์มอัจฉริยะนั้นตั้งอยู่บนแนวคิดของการทำเกษตรสมัยใหม่ที่เรียกว่า เกษตรแม่นยำสูง การทำการเกษตรที่เป็นมิตรกับสิ่งแวดล้อม โดยทำให้เกษตรกรสามารถปรับการใช้ทรัพยากรให้สอดคล้องกับสภาพของพื้นที่มากที่สุด รวมไปถึงเรื่องการดูแลอย่างมีประสิทธิภาพ

ผู้จัดทำจึงมีแนวคิดที่จะนำเอาเทคโนโลยีไอโอที (IoT) มาใช้ในการควบคุมอุณหภูมิและความชื้นในโรงเพาะเห็ดนางฟ้าภูฐาน โดยจะนำข้อมูลที่เก็บค่าได้ส่งข้อมูลผ่านคลาวด์ (Cloud) และแสดงข้อมูลบนแอปพลิเคชัน มาช่วยในการควบคุมให้สามารถรับรู้ ค่าอุณหภูมิและความชื้นของสภาพแวดล้อมภายในโรงเพาะเห็ดนางฟ้าภูฐานเพื่อเป็นการลดสภาพความแปรปรวน ของดินฟ้าอากาศและเพื่อเพิ่มผลผลิตที่มากขึ้นโดยที่ค่าของอุณหภูมิและความชื้นภายในโรงเพาะเห็ดนางฟ้าภูฐาน

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อพัฒนาระบบการควบคุมและการจัดการโรงเพาะเห็ดนางฟ้าภูฐานผ่านอุปกรณ์พกพา

1.2.2 เพื่อนำเทคโนโลยี IoT มาประยุกต์ใช้กับการเพาะเห็ดนางฟ้าภูฐาน

1.3 ขอบเขตของการวิจัย

1.3.1 ระบบการควบคุมและการจัดการโรงเพาะเห็ดนางฟ้าภูฐานผ่านสมาร์ทโฟน (Smart phone) จะทำการใช้บอร์ด Arduino Uno ในการควบคุมเซนเซอร์ภายในโรงเพาะเห็ด โดยจะสามารถควบคุมสภาพความชื้นอุณหภูมิ และแสง ให้อยู่ในค่าที่เหมาะสมกับการเจริญเติบโตของเห็ด

1.3.2 ขอบเขตของฮาร์ดแวร์

1.3.2.1 สามารถวัดอุณหภูมิและความชื้นภายในโรงเพาะเห็ดได้ผ่านเซนเซอร์วัดอุณหภูมิ

1.3.2.2 สามารถระบายความร้อนเพื่อลดอุณหภูมิได้ผ่านพัดลม

1.3.2.3 สามารถเพิ่มความชื้นได้โดยการรดน้ำผ่านสปริงเกอร์

1.3.2.4 สามารถวัดคุณภาพอากาศได้โดยจะสามารถวัดก๊าซที่เป็นอันตรายต่อมนุษย์ได้

1.3.3 ขอบเขตของแอปพลิเคชัน

1.3.3.1 แอปพลิเคชันมีระบบสมาชิก

1.3.3.2 แอปพลิเคชันสามารถดูค่าความชื้นและอุณหภูมิในโรงเพาะเห็ดได้

1.3.3.3 แอปพลิเคชันสามารถดูการเจริญเติบโตของเห็ดได้ผ่านกล้อง

1.3.3.4 แอปพลิเคชันสามารถดูค่าสภาพก๊าซในอากาศได้

1.3.3.5 แอปพลิเคชันสามารถแจ้งเตือนได้

1.3.4 ขอบเขตของผู้ใช้งาน

1.3.4.1 สามารถทำการสมัครสมาชิกผ่านบนสมาร์ตโฟนได้

1.3.4.2 สามารถทำการควบคุมระบบของโรงเพาะเห็ดได้ผ่านสมาร์ตโฟน

1.3.4.3 สามารถตรวจสอบอุณหภูมิและสภาพความชื้นได้ผ่านสมาร์ตโฟน

1.4 เครื่องมือที่ใช้ในงานวิจัย

1.4.1 ระบบปฏิบัติการ

- Windows

1.4.2 เครื่องมือสำหรับพัฒนาโปรแกรม

- Arduino IDE version 1.8.19
- Android Studio 2022.1
- Visual Studio code 1.77.3

1.4.3 โปรแกรมฐานข้อมูล

- Firebase

1.4.4 ภาษาคอมพิวเตอรืที่ใช้พัฒนาโปรแกรม

- C/C++
- Dart
- Flutter

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ใช้ระบบที่พัฒนาเพื่อมาช่วยอำนวยความสะดวกให้กับผู้ที่ทำการเพาะเห็ด

1.5.2 ใช้เป็นแนวทางในการพัฒนาเกษตรกรให้มีความรู้ความสามารถในการนำเทคโนโลยีมาประยุกต์ใช้กับการเกษตร

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในงานวิจัยครั้งนี้ ผู้ศึกษางานวิจัยจะอธิบายเกี่ยวกับเครื่องมือและอุปกรณ์ต่างๆที่เราจะนำมาประยุกต์ใช้ รวมถึงขั้นตอนการทำงานของเทคโนโลยี IOT ที่ผู้วิจัยได้ศึกษามา ในภาคเกษตรกรรมให้เกิดประโยชน์สูงสุด

2.1 ทฤษฎีที่เกี่ยวข้อง

ระบบฝังตัว หรือ สมองกลฝังตัว (embedded system) คือระบบประมวลผล ที่ใช้ชิปหรือไมโครโพรเซสเซอร์ที่ออกแบบมาโดยเฉพาะ เป็นระบบคอมพิวเตอร์ขนาดเล็กที่ฝังไว้ในอุปกรณ์เครื่องใช้ไฟฟ้า และเครื่องเล่นอิเล็กทรอนิกส์ต่างๆ เพื่อเพิ่มความฉลาด ความสามารถให้กับอุปกรณ์เหล่านั้นผ่านซอฟต์แวร์ซึ่งต่างจากระบบประมวลผลที่เครื่องคอมพิวเตอร์ทั่วไป ระบบฝังตัวถูกนำมาใช้กันอย่างแพร่หลายในยานพาหนะ เครื่องใช้ไฟฟ้าในบ้านและสำนักงาน อุปกรณ์อิเล็กทรอนิกส์ เทคโนโลยีซอฟต์แวร์ เทคโนโลยีฮาร์ดแวร์ เทคโนโลยีเครือข่ายเน็ตเวิร์ก เทคโนโลยีด้านการสื่อสาร เทคโนโลยีเครื่องกลและของเล่นต่าง ๆ คำว่าระบบฝังตัวเกิดจากการ ที่ระบบนี้เป็นระบบประมวลผลเช่นเดียวกับระบบคอมพิวเตอร์ แต่ว่าระบบนี้จะฝังตัวลงในอุปกรณ์อื่น ๆ ที่ไม่ใช่เครื่องคอมพิวเตอร์ ในปัจจุบันระบบสมองกลฝังตัวได้มีการพัฒนามากขึ้น โดยในระบบสมองกลฝังตัวอาจจะประกอบไปด้วยไมโครคอนโทรลเลอร์ หรือ ไมโครโพรเซสเซอร์ อุปกรณ์ที่ใช้ระบบสมองกลฝังตัวที่เห็นได้ชัดเช่น โทรศัพท์มือถือ และในระบบสมองกลฝังตัวยังมีการใส่ระบบปฏิบัติการต่างๆแตกต่างกันไปอีกด้วย ดังนั้น ระบบสมองกลฝังตัวอาจจะทำงานได้ตั้งแต่ควบคุมหลอดไฟจนไปถึงใช้ในยานอวกาศ

Smart Farm หรือเกษตรอัจฉริยะ คือการทำการเกษตรที่นำเอาระบบเทคโนโลยีเข้ามาช่วยในการบริหารจัดการดูแลการเพาะปลูก รวมไปถึงกระบวนการผลิตเพื่อนำไปสู่การเกษตรเชิงธุรกิจ ด้วยการผสมผสานศาสตร์ทางวิทยาศาสตร์การเกษตร กับศาสตร์ทางวิศวกรรมเทคโนโลยี อิเล็กทรอนิกส์ และนำเอาเทคโนโลยีสมัยใหม่เข้ามาผสมผสานเข้ากับงานด้านการเกษตรเพื่อช่วยแก้ไขปัญหาต่างๆ ให้กับเกษตรกร

IoT หรือ Internet of Things (อินเทอร์เน็ตของสรรพสิ่ง) หมายถึง วัตถุ อุปกรณ์ พาหนะ สิ่งของเครื่องใช้ และสิ่งอำนวยความสะดวกในชีวิตอื่น ๆ ที่มนุษย์สร้างขึ้นโดยมีการฝังตัวของวงจร อิเล็กทรอนิกส์ ซอฟต์แวร์ เซนเซอร์ และการเชื่อมต่อกับเครือข่าย ซึ่งวัตถุสิ่งของเหล่านี้ สามารถเก็บ บันทึกและแลกเปลี่ยนข้อมูลกันได้ อีกทั้ง สามารถรับรู้สภาพแวดล้อมและถูกควบคุมได้จากระยะไกล

ผ่านโครงสร้างพื้นฐานการเชื่อมต่อเข้ากับสมาร์ทโฟนเท่านั้น แต่ IoT สามารถประยุกต์ใช้กับอุปกรณ์ทุกอย่างที่ถูกรออกแบบมาให้เชื่อมโยงกันได้บนเครือข่ายอินเทอร์เน็ตเพื่อที่จะสามารถสื่อสารกันได้

เซนเซอร์วัดความชื้นและอุณหภูมิ DHT11 แบบติดตั้งลงแผ่น PCB ง่ายต่อการใช้งาน โมดูลวัดอุณหภูมิและความชื้น (DHT11 Temperature and Humidity Sensor Module) เป็นโมดูลที่สามารถวัดอุณหภูมิและความชื้นบริเวณรอบๆทั่วไปหรือในห้องหรือประยุกต์ใช้งานอื่น เช่น Testing, Inspection Equipment, Automatic Control, Data Logger, Weather Station, Humidity Regulator ขึ้นอยู่กับการเขียนโปรแกรมและการต่อใช้งานภายนอก ผ่านการสอบเทียบตามมาตรฐาน มีความน่าเชื่อถือ ราคาถูก การตอบสนองที่รวดเร็ว และความแม่นยำในการวัดค่าสูงสามารถใช้งานกับไมโครคอนโทรลเลอร์ทั่วไปได้ ในโมดูลประกอบไปด้วย ส่วนวัดความชื้นแบบ Resistive type และส่วนวัดอุณหภูมิแบบNTC ให้สัญญาณเอาต์พุตแบบ Digital Output, การตรวจวัดคงที่กับ DHT11 Sensor, ย่านวัดอุณหภูมิ 0-50 องศาเซลเซียส, ย่านวัดความชื้น 20-90% RH, ง่ายต่อการติดตั้งใช้งาน

แอลดีอาร์ (LDR) หรือชื่อเต็ม ๆ คือ Light Dependent Resistor หรือตัวต้านทานที่แปรค่าตามแสง คือ ตัวต้านทานชนิดที่เปลี่ยนสภาพความนำไฟฟ้า (Conductance) ได้เมื่อมีแสงมาตกกระทบ ทำจากวัสดุสารกึ่งตัวนำที่ไวต่อแสง บางครั้งเราเรียก LDR เซนเซอร์ชนิดนี้ว่าโฟโตริซิสเตอร์ (Photoresistor) หรือ โฟโตคอนดักเตอร์ (Photoconductor) โมดูลนี้ให้สัญญาณเอาต์พุตได้ ทั้งแบบแอนะล็อกที่ช่อง (AO) ซึ่งมีค่าระหว่าง 0 - 1023 และแบบดิจิทัลที่ช่อง (DO) ค่า 0 กับ 1 โดยสามารถปรับระดับแรงดันที่นำไปเปรียบเทียบได้โดยการหมุนตัวต้านทานปรับค่าได้ (VR) บนบอร์ด และจะต้องป้อนใช้ไฟเลี้ยง 3.3-5V ให้กับวงจร ซึ่งบนบอร์ดจะมีแอลอีดีแสดงสัญญาณไฟเลี้ยง (PWR LED) และระดับสัญญาณที่เปรียบเทียบ (DO LED) ด้วย

MQ 135 Air quality เป็นเซนเซอร์ตรวจวัดคุณภาพอากาศ โดยเซนเซอร์จะตรวจวัดแก๊สที่เป็นอันตรายต่อร่างกายมนุษย์ เช่นแก๊สในกลุ่ม NH₃, NO_x, alcohol, Benzene, CO, CO₂ ทำให้ถูกนำมาใช้ในงานควบคุมสภาพอากาศ, ควบคุมการทำงานของเครื่องฟอกอากาศ เป็นต้น เมื่อป้อนแรงดันให้แก่เซนเซอร์ ต้องรอการปรับค่าสภาพอากาศเริ่มต้น 20 วินาที ก่อนทำการวัด ใช้ไฟเลี้ยง 5V ESP8266 คือ Wifi ไมโครชิปขนาดเล็ก หรือจะเรียกว่าเป็นสมองกลแบบฝังตัวชนิดต่อ wifi ได้ มีราคาถูก ผลิตโดย Espressif Systems ถูกใช้อย่างแพร่หลายในอุปกรณ์ IOT และ อุปกรณ์สมาร์ทโฮม ต่างๆ ถึงแม้ว่าจะมีการพัฒนารุ่นใหม่เช่น ESP32 ออกมา แต่ ESP8266 ก็ยังได้รับความนิยมเป็นอย่างสูงอยู่ NodeMCU ESP8266 บอร์ดยอดนิยมนี้ใช้โมดูล ESP-12E และมาพร้อมกับ Flash Memory 4 Mbits พร้อมกับอินเทอร์เฟซการสื่อสาร 4 แบบ คือ SPI, I2C, UART และ I2S พร้อม Digital I/O pin 16 ขา และ Analog Input pin (ADC) 1 ขา RAM 160KB แบ่งออกเป็น 64KB สำหรับคำสั่งและ 96KB สำหรับข้อมูล ที่มีขายในท้องตลาดจะมีรุ่น V2, V3 และยังมีการพัฒนาเวอร์

ชั้นใหม่ๆออกมาเรื่อยๆ การ flash firmware และจ่ายไฟเลี้ยงทำได้ง่ายโดยผ่าน port mini USB บนบอร์ด

คาร์บอนไดออกไซด์เป็นแก๊สที่มีอยู่ทั่วไปในบรรยากาศ แก๊สนี้มีบทบาทสำคัญต่อการดำรงชีพของทั้งมนุษย์ สัตว์ และพืช เป็นสารตั้งต้นที่พืชใช้ผลิตอาหารโดยกระบวนการสังเคราะห์แสง ในด้านอุตสาหกรรมนั้น คาร์บอนไดออกไซด์ถูกนำมาใช้ประโยชน์ในหลายๆ ด้าน โอกาสการเกิดพิษของแก๊สชนิดนี้ ในการทำงานโดยปกติมีโอกาสเกิดขึ้นได้น้อย อย่างไรก็ตามหากได้รับแก๊สนี้เข้าไปในปริมาณมาก จะทำให้หายใจเร็ว ชีพจรเร็ว หัวใจเต้นเร็ว กดสมอง ซึม มึนงง สับสน หมดสติ และอาจเสียชีวิตได้ แก๊สคาร์บอนไดออกไซด์ก่อโรคได้ทางหนึ่งโดยการแทนที่ออกซิเจน (Asphyxiant) ทำให้ออกซิเจนในอากาศมีไม่พอ จึงเกิดพิษจากภาวะออกซิเจนในเลือดต่ำ (Hypoxia) ขึ้นได้ การที่มีคาร์บอนไดออกไซด์คั่งในเลือด (Hypercapnia) ไม่ว่าจะจากการขาดออกซิเจนหรือได้รับคาร์บอนไดออกไซด์เข้าไปมากก็ตาม จะทำให้เลือดเป็นกรด (Acidosis) เกิดการขยายตัวของหลอดเลือด กระตุ้นระบบหายใจให้หายใจเร็วขึ้น ทำให้หัวใจเต้นเร็ว และกดสมองอาการเฉียบพลัน การได้รับแก๊สคาร์บอนไดออกไซด์นั้น เข้าสู่ร่างกายโดยการสูดหายใจเข้าไปเป็นหลัก การสัมผัสกับแก๊สที่ผิวหนังหรือกลืนกินเข้าไปไม่ทำให้เกิดพิษ เมื่อสูดหายใจเอาแก๊สเข้าไป ในระยะแรกจะทำให้เกิดอาการหายใจเร็ว หายใจลึกขึ้น ความดันโลหิตสูงขึ้น หัวใจเต้นเร็ว ชีพจรเร็ว หากได้รับในปริมาณมากขึ้น จะเริ่มมีผลกดสมอง ทำให้ซึมลง ปวดศีรษะ วิงเวียนศีรษะ มึนงง สับสน การได้ยินลดลง และรบกวนการมองเห็น เนื่องจากสมองถูกกดการทำงาน ที่ผิวหนังจะเกิดหลอดเลือดขยายตัว เหงื่อออก กล้ามเนื้อสั่นกระตุก (Tremor) อาจพบมีคลื่นไส้ อาเจียน และท้องเสียได้ บางรายอาจมีอาการคลั่ง (Panic) หากได้รับปริมาณสูงมากจะทำให้หมดสติ และเสียชีวิตในที่สุด อาการพิษจากแก๊สคาร์บอนไดออกไซด์นี้ มักจะพบร่วมกับภาวะขาดออกซิเจน (Hypoxia) ได้เสมอ ซึ่งภาวะขาดออกซิเจน อาจนำไปสู่อาการอื่นๆ เช่น สมองตาย ไตเสื่อม ตาบอด ตามมาได้ ในผู้ป่วยที่รอดชีวิตจากเหตุการณ์ที่ทะเลสาบนีออส [10] หลายรายมีอาการ ไอ ไอเป็นเลือด หอบเหนื่อย ระคายเคืองตา และแผลไหม้ที่ผิวหนังด้วย อย่างไรก็ตามเนื่องจากแก๊สคาร์บอนไดออกไซด์ที่รั่วไหลในเหตุการณ์ทะเลสาบนีออสนี้มีไอความร้อนจากภูเขาไฟปะปนมาด้วย อาการระคายเคืองและแผลไหม้ที่ผิวหนังดังกล่าว จึงอาจจะเกิดจากไอความร้อน ไม่ได้เกิดจากพิษของแก๊สคาร์บอนไดออกไซด์ก็ได้

2.2 งานวิจัยที่เกี่ยวข้อง

นายอะหมัด วันตา (2563) แนวคิดหลักของสมาร์ทฟาร์ม คือ การประยุกต์ใช้เทคโนโลยี อิเล็กทรอนิกส์และคอมพิวเตอร์ รวมถึงเทคโนโลยีสารสนเทศและการสื่อสาร ในการพัฒนาทั้งห่วงโซ่ อุปทานหรือเครือข่ายโลจิสติกส์ของ กระบวนการผลิตสินค้าเกษตรไปจนถึงผู้บริโภค เพื่อยกระดับ คุณภาพการผลิต ลดต้นทุน รวมทั้งพัฒนา มาตรฐานสินค้า ในปัจจุบันได้มีการนำเทคโนโลยีที่เรียกว่า อินเทอร์เน็ตของทุกสรรพสิ่ง มาใช้ในสมาร์ท ฟาร์มอย่างแพร่หลาย ยกตัวอย่างเช่น การนำ IoT มาใช้ ในการควบคุมระบบพลังงานหมุนเวียนการรดน้ำ การคาดคะเนความต้องการพลังงานแสงอาทิตย์และ ติดตั้งเซลล์พลังงานแสงอาทิตย์ขนาดที่เหมาะสม มี การนำอัลกอริทึมตรรกะคลุมเครือมาวิเคราะห์ สภาพแวดล้อมและดิน เพื่อตัดสินใจว่าเมื่อใดควรรดน้ำใน ฟาร์ม เกษตรกรสามารถใช้งานระบบได้ใน รูปแบบที่หลากหลาย สามารถตรวจสอบการทำงานได้จาก ระยะเวลา

วีรศักดิ์ ฟองเงิน สุรพงษ์ เพ็ชรหาญ และ รัฐสิทธิ์ ยะจ่อ (2561) ในปัจจุบันเห็ดนางฟ้าถือเป็นอาหารที่ได้รับความนิยมเป็นอย่างมากเนื่องจากมีประโยชน์และให้คุณค่า ทางอาหารสูงจึงทำให้ เกษตรกรหันมาทำอาชีพเพาะเห็ดเพิ่มขึ้นบางกลุ่มก็ประสบผลสำเร็จบางกลุ่มก็ล้มเหลวซึ่งเกิด จาก ปัจจัยหลายๆด้านดังนั้นจึงต้องมีการศึกษาและวางแผนเป็นอย่างดีจากสภาพแวดล้อมและภูมิอากาศที่ เปลี่ยนแปลงในแต่ละฤดูทำให้ส่งผลกระทบต่อกลุ่มอาชีพเพาะเห็ดโดยเฉพาะอุณหภูมิซึ่งมีผลต่อการ เจริญเติบโตของ ดอกเห็ดโครงการวิจัยนี้จึงได้ออกแบบระบบควบคุมอุณหภูมิและความชื้นสำหรับ โรงเรือนเพาะเห็ดด้วย ไมโครคอนโทรลเลอร์โดยมีการศึกษาค้นคว้าข้อมูลที่เกี่ยวข้องต่อการเพาะเห็ด ตลอดจนการออกแบบโครงสร้าง โรงเรือนที่เหมาะสมโดยแบ่งการทดสอบออก 2 ส่วนคือการทดสอบ ในส่วนของระบบควบคุมและการทดสอบผลผลิต ของดอกเห็ดในโรงเรือนที่มีการควบคุมอุณหภูมิและ ความชื้นโดยนำก้อนเห็ดนางรมและเห็ดนางฟ้ามาทดสอบจำนวน 300 ก้อนและเปรียบเทียบ ประสิทธิภาพโรงเรือนเพาะเห็ดที่มีการควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นกับโรงเรือน ผลจากการ ออกแบบและพัฒนาเครื่องควบคุมอุณหภูมิความชื้นในโรงเพาะเห็ดที่ได้จัดทำขึ้นนี้สามารถควบคุม อุณหภูมิและความชื้นโดยทำการปรับตั้งค่าทั้งสองจากชุดควบคุมซึ่งสามารถปรับตั้งค่าได้ตามการใช้ งานจริงเพื่อให้ตรงตามความต้องการของเห็ดแต่ละชนิดทำให้ผลผลิตเห็ดที่ได้ออกอย่างสม่ำเสมอ สามารถ ประหยัดเวลาในการให้น้ำในโรงเพาะเห็ดและทำงานสัมพันธ์ร่วมกับพัดลมระบายอากาศที่ ติดตั้งภายในโรงเพาะเห็ด ทำให้อากาศในโรงเพาะเห็ดเกิดการถ่ายเทอุณหภูมิโดยรอบได้อย่างรวดเร็ว

โดยอาศัยการตรวจวัดอุณหภูมิและ ความชื้นจากอุปกรณ์ที่ติดตั้งไว้ภายในโรงเพาะเห็ดสามารถนำไปประยุกต์ใช้กับโรงเพาะเห็ดอื่น ๆ ได้

นายศิวกกร จินดารัตน์ (2557) งานวิจัยมีวัตถุประสงค์เพื่อพัฒนาระบบจัดการฟาร์มอัจฉริยะ ด้วยระบบอินเทอร์เน็ตและอาตุน้อย และพัฒนาระบบการควบคุมและจัดการฟาร์มอัจฉริยะผ่านทางสมาร์ตโฟน ผลการวิจัยพบว่า การสร้างระบบอัจฉริยะ โดยนำเอา Embedded System และ Smart Phone เข้ามาผสมผสานเข้ากับการ จัดการฟาร์มเลี้ยงไก่ เข้ามาช่วยแก้ไขปัญหาให้กับเกษตรกรฟาร์มไก่ โดยใช้ Raspberry Pi และ Arduino Uno งานวิจัยนี้ได้ทำการทดลองและวิเคราะห์ผลและเปรียบเทียบการนำระบบอัจฉริยะมา ประยุกต์ใช้งานกับสถานที่ตัวอย่างเพื่อเพิ่มประสิทธิภาพมากยิ่งขึ้น ที่สามารถรับรู้ ติดตาม สภาพอากาศความชื้นอุณหภูมิ,คุณภาพอากาศและควบคุมการเปิด/ปิดไฟกับพัดลมในฟาร์มเลี้ยง ไก่ เป็นการควบคุมและจัดการฟาร์มไก่อัจฉริยะได้ทุกที่ทุกเวลา เพิ่มความสะดวกในการบริหาร จัดการฟาร์มได้อย่างมีประสิทธิภาพและประหยัด ค่าใช้จ่ายและช่วยลดต้นทุน ของเกษตรกรผู้เลี้ยงไก่ต่อไป

บทที่ 3

วิธีดำเนินงานวิจัย

วิธีการดำเนินงานและขั้นตอนการพัฒนาโรงเพาะเห็ดนางฟ้าภูฐานระบบปิดผ่านสมาร์ทโฟน
ผู้วิจัยได้วางแผนการดำเนินการมีขั้นตอน ดังนี้

ระยะเวลาในการดำเนินงาน

ตารางที่ 3-1 GANTT CHART แสดงระยะเวลาในการดำเนินการ

กิจกรรม	พ.ศ. 2565												พ.ศ. 2566		
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.ศึกษาและเก็บรวบรวมข้อมูล															
2.เสนอหัวข้อโครงการ															
3.พัฒนาแอปพลิเคชัน															
4.ส่งข้อมูลโปรแกรม															
5.พัฒนาระบบในส่วนฮาร์ดแวร์															
6.ทดสอบระบบและอุปกรณ์ฮาร์ดแวร์															
7.แก้ไขและปรับปรุงข้อผิดพลาดของระบบและฮาร์ดแวร์															
8.สรุปผลและนำเสนอ															
9.จัดทำเอกสาร															

3.1 ศึกษาและเก็บรวบรวมข้อมูล

3.1.1 ศึกษาเกี่ยวกับการเจริญเติบโตของเห็ดนางฟ้าภูฐานและวิธีการเพาะเห็ด

ผู้วิจัยได้ทำการศึกษาถึงสภาพแวดล้อมที่เหมาะสมและปัจจัยในการเพาะเห็ด จึงทำให้ทราบปัจจัยในการเพาะนั้นประกอบไปด้วย อุณหภูมิ สภาพความชื้น อากาศ และแสง

3.1.2 ศึกษาปัญหาและวิเคราะห์ความต้องการของระบบ

ผู้วิจัยได้ทำการศึกษาถึงสภาพปัญหาของการเพาะเห็ดของเกษตรกรว่ามีความไม่สะดวกในด้านใดและได้ทำการวิเคราะห์ถึงระบบที่ผู้วิจัยจะจัดทำขึ้นว่าสามารถนำเทคโนโลยีเข้ามาช่วยในด้านไหนได้บ้าง

3.1.3 ศึกษาอุปกรณ์เครื่องมือต่างๆที่ใช้ในการพัฒนาระบบ

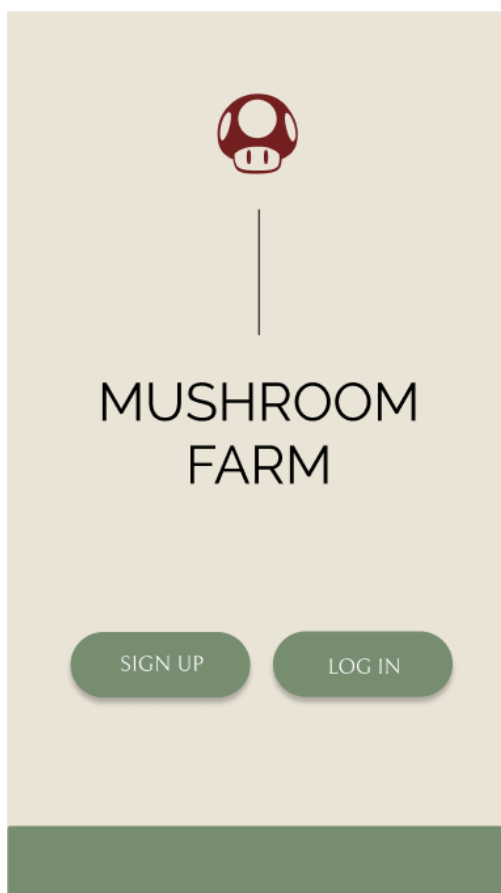
3.1.3.1 ซอฟต์แวร์ในการเขียนโปรแกรมควบคุมบอร์ด ESP8266

ในงานวิจัยนี้จะทำการใช้บอร์ด ESP8266 เป็นบอร์ดที่มีการติดต่อสื่อสารผ่านทางสาย USB โดยการใช้บอร์ดจะมีการแปลงการสื่อสาร USB เป็น COM พอร์ตในตัว ดังนั้นก่อนที่จะใช้งานจะต้องมีการติดตั้งไดรเวอร์ก่อน

การใช้งานบอร์ด ESP8266 จะเขียนชุดคำสั่งบน Arduino IDE และจะใช้
ภาษา C/C++ ในการเขียนชุดคำสั่ง

- 3.1.3.2 เซนเซอร์วัดอุณหภูมิและความชื้น
- 3.1.3.3 เซนเซอร์วัดคุณภาพอากาศ
- 3.1.3.4 การต่อวงจรของพัดลมระบายอากาศ
- 3.1.3.5. การต่อวงจรการทำงานของปั้มน้ำ

3.2 Mock UP ของแอปพลิเคชัน

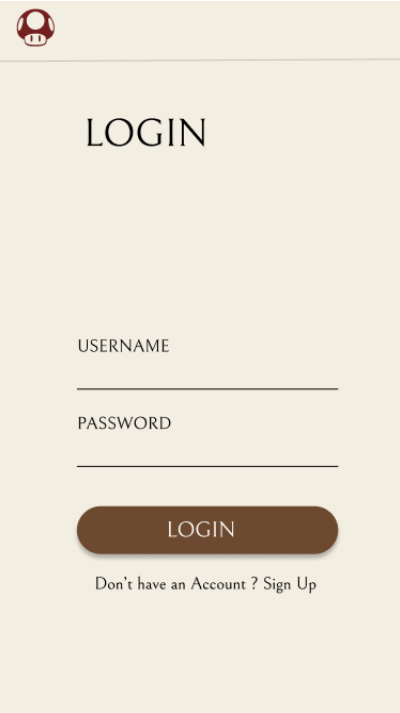


ภาพที่ 3-1 หน้าแรกของแอป เพื่อให้เลือกที่จะ
SIGN UP(ลงทะเบียน) หรือ LOG IN (เข้าสู่ระบบ)



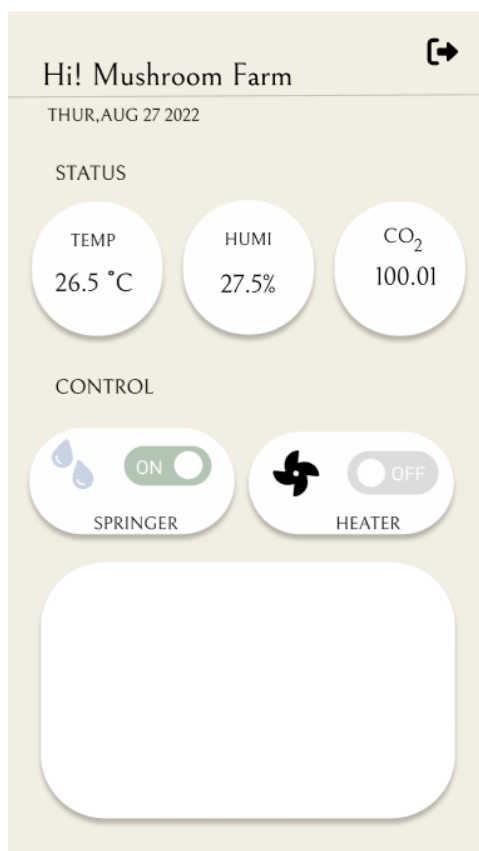
The image shows a 'SIGN UP' form on a light beige background. At the top left is a small red circular logo with a white 'H'. The title 'SIGN UP' is centered in a large, black, serif font. Below the title are two input fields: 'USERNAME' and 'PASSWORD', each with a horizontal line underneath. A dark brown, rounded rectangular button with the text 'SIGNUP' in white is centered below the fields. At the bottom, the text 'Already have Account ? LOGIN' is centered in a small, black, sans-serif font.

ภาพที่ 3-2 หน้า SIGN UP(ลงทะเบียน)
สำหรับคนที่ยังไม่เคยเข้าใช้ระบบ



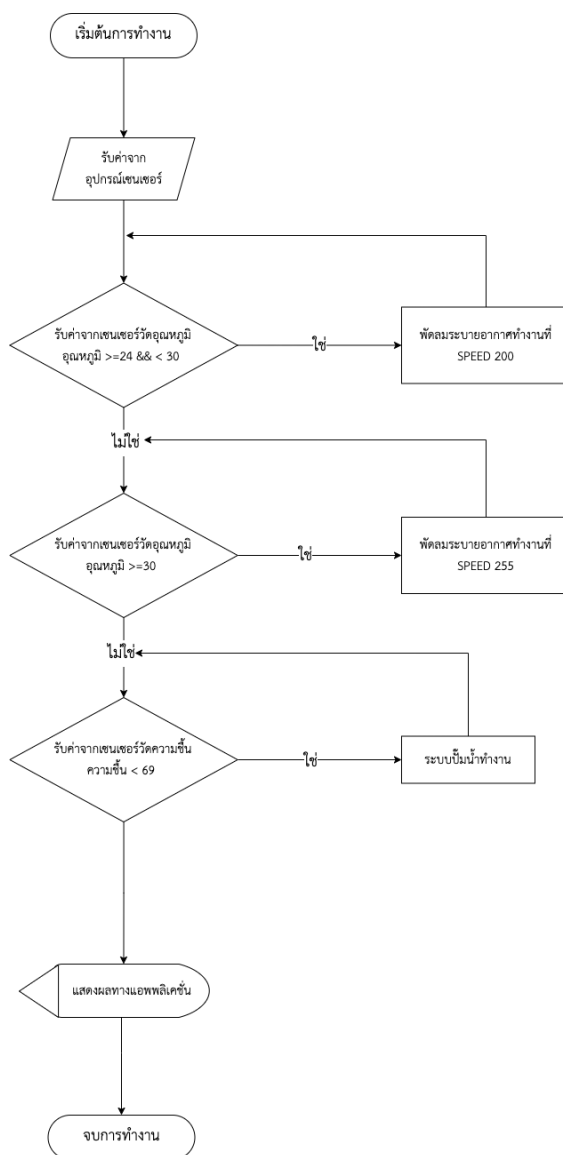
The image shows a 'LOGIN' form on a light beige background. At the top left is a small red circular logo with a white 'H'. The title 'LOGIN' is centered in a large, black, serif font. Below the title are two input fields: 'USERNAME' and 'PASSWORD', each with a horizontal line underneath. A dark brown, rounded rectangular button with the text 'LOGIN' in white is centered below the fields. At the bottom, the text 'Don't have an Account ? Sign Up' is centered in a small, black, sans-serif font.

ภาพที่ 3-3 LOG IN (เข้าสู่ระบบ) เพื่อ
เข้าใช้งานแอปควบคุมโรงเพาะเห็ด



ภาพที่ 3-4 เมื่อเข้าสู่ระบบจะพบหน้าควบคุมค่าต่างๆภายในโรงเรือนดังนี้

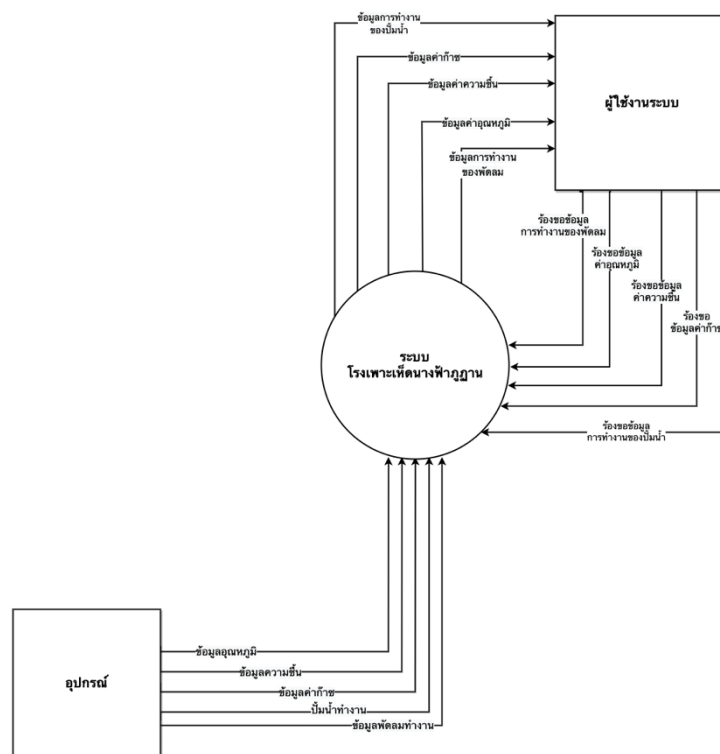
3.2.3 Flow chart การทำงานของระบบโรงเพาะเห็ดนางฟ้าภูฐาน



ภาพที่ 3-5 FLOWCHART แสดงขั้นตอนการทำงานของระบบโรงเพาะเห็ดนางฟ้าภูฐาน

อธิบายขั้นตอนการทำงานของ Flowchart จะเริ่มต้นที่อุปกรณ์เซนเซอร์ตรวจจับวัดค่าอุณหภูมิและค่าความชื้นสัมพัทธ์จะทำการตรวจจับค่าที่วัดได้ภายในโรงเพาะเห็ดนางฟ้าภูฐานถ้าอุณหภูมิภายในโรงเพาะเห็ดสูงกว่า 30 องศาเซลเซียสและค่าความชื้นน้อยกว่า 70% ระบบพัฒนาระบายอากาศและปั๊มน้ำจะทำงานอัตโนมัติ

3.2.4 การออกแบบ Context Diagram ของระบบโรงเพาะเห็ดนางฟ้าภูฐาน



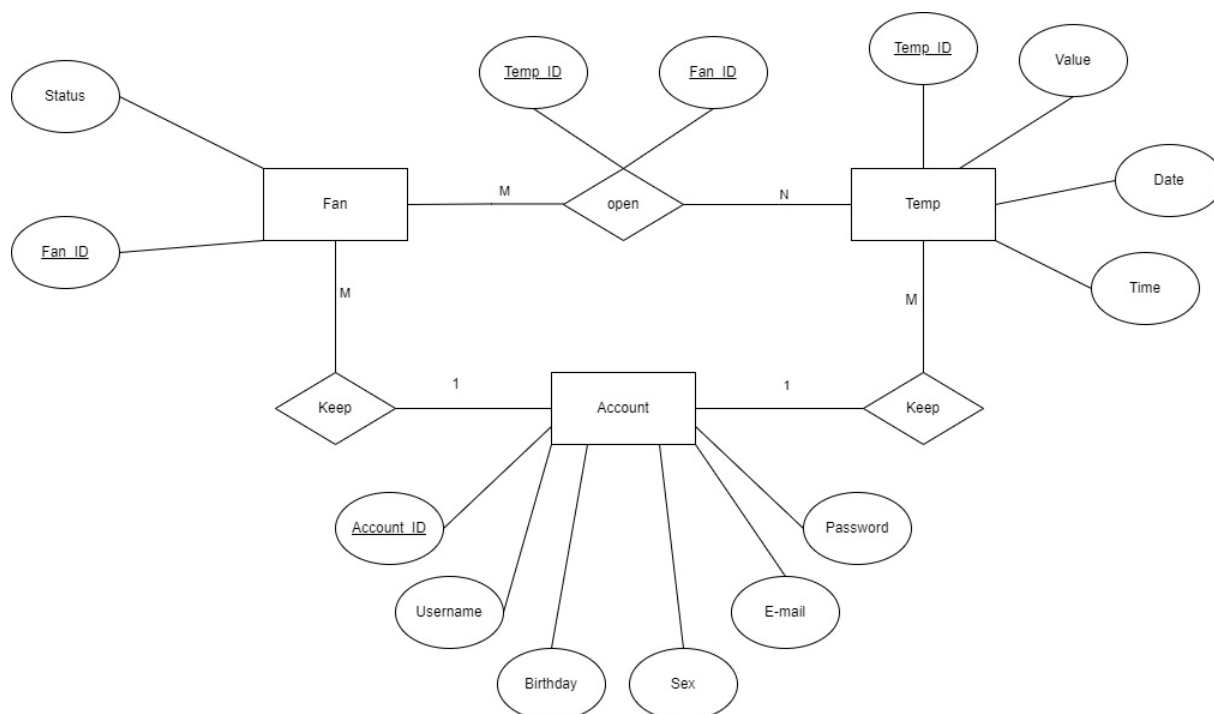
ภาพที่ 3-6 CONTEXT DIAGRAM แสดงการทำงานของระบบโรงเพาะเห็ดนางฟ้าภูฐาน

คำอธิบาย Context Diagram ของระบบโรงเพาะเห็ดนางฟ้าภูฐาน

ระบบโรงเพาะเห็ดนางฟ้าภูฐาน จะประกอบไปด้วย External Entity ที่เกี่ยวข้องกัน 2 ประเภท คือ ผู้ใช้งานระบบและอุปกรณ์

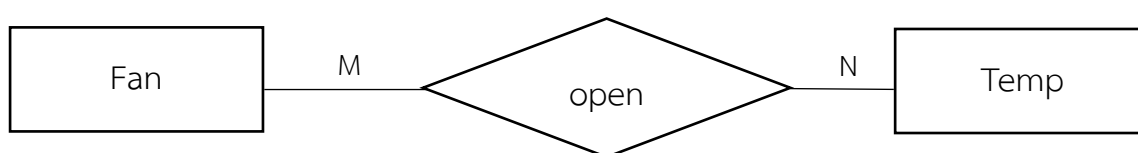
1. ผู้ใช้งานระบบสามารถทำการร้องขอข้อมูลค่าอุณหภูมิ ข้อมูลค่าความชื้น ข้อมูลค่าก๊าซ และสามารถสั่งการทำงานของพัดลมระบายอากาศได้และระบบก็จะทำการส่งข้อมูลค่าอุณหภูมิ ข้อมูลค่าความชื้น ข้อมูลค่าก๊าซ สถานะการทำงานของพัดลม กลับไปให้ผู้ใช้งานระบบ
2. อุปกรณ์เซนเซอร์ต่างๆจะทำการส่งมูลค่าอุณหภูมิ ข้อมูลค่าความชื้น ข้อมูลค่าก๊าซ และสามารถสั่งการทำงานของพัดลมระบายอากาศไปยังระบบโรงเพาะเห็ดนางฟ้าภูฐาน

3.2.7 ER-Diagram Model



ภาพที่ 3-9 ER-DIAGRAM MODEL แสดงแบบจำลองโครงสร้างของแอปพลิเคชัน

ภาพความสัมพันธ์และคำอธิบาย



พัดลมหลายตัวสามารถเปิดหรือปิดได้ในอุณหภูมิหลายช่วงขณะ



พัดลมหลายตัวเก็บข้อมูลไว้ใน 1 บัญชี



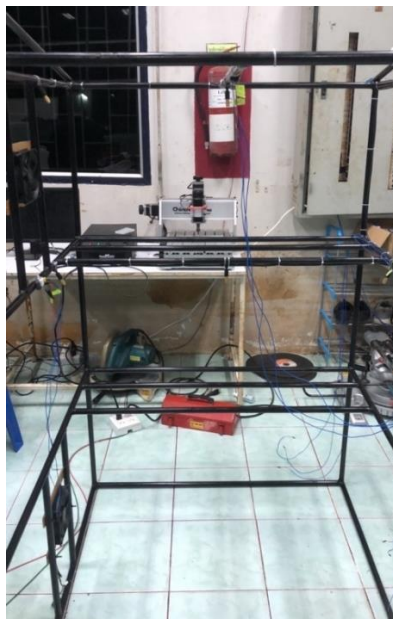
อุณหภูมิหลายช่วงขณะจะถูกเก็บไว้ใน 1 บัญชี

3.3 การพัฒนาในส่วนของฮาร์ดแวร์



ภาพที่ 3-10 โครงของโรงเรือนสำหรับเพาะเห็ด

โครงสร้างใช้เหล็กเส้นกลมมาตัดและเชื่อมกันโดยมีขนาดกว้าง 1 เมตร ยาว 1 เมตร ส่วนฝั่งข้างหน้าสูง 1.70 เมตร ฝั่งหลังสูง 1.50 เมตร และเชื่อมเหล็กเป็นเส้นคู่กันแบ่งเป็น 2 ชั้นดังภาพที่ 3-10



ภาพที่ 3-11 ตำแหน่งของการติดตั้งเซนเซอร์ภายในโรงเรือน

ผู้จัดทำได้ทำการติดตั้งเซนเซอร์วัดอุณหภูมิและความชื้นจำนวน 1 ตัวติดตั้งไว้บริเวณโครงด้านข้างของโรงเรือนและได้ทำการติดตั้งเซนเซอร์วัดคุณภาพอากาศจำนวน 1 ตัว ไว้บริเวณคานด้านบนของโรงเรือนดังภาพที่ 3-11



ภาพที่ 3-12 โครงของโรงเรือนเมื่อคลุมสแลนแล้ว

ผู้จัดทำจะทำการนำสแลนมาคลุมโรงเรือนเพื่อใช้บังแสงที่จะเข้ามาภายในของโรงเรือน เนื่องจากการเพาะเห็ดนั้นจำเป็นที่จะต้องมีการควบคุมความสว่างภายในโรงเรือน ดังภาพที่ 3-12



ภาพที่ 3-13 โรงเรือนเมื่อคลุมพลาสติก

ผู้จัดทำจะทำการนำพลาสติกมาคลุมตัวโรงเรือนจำนวน 1 ชั้น เพื่อเป็นการทำให้ภายในโรงเรือนเป็นระบบปิด ดังภาพที่ 3-13



ภาพที่ 3-14 ตำแหน่งของการติดตั้งพัดลมระบายอากาศและปั้มน้ำ

ตำแหน่งของการติดตั้งพัดลมระบายอากาศและปั้มน้ำจะทำการติดตั้งไว้บริเวณด้านข้างของโรงเรือน เพาะเห็ด ดังภาพที่ 3-1



ภาพที่ 3-15 ตำแหน่งของการติดตั้งหัวพ่นน้ำ

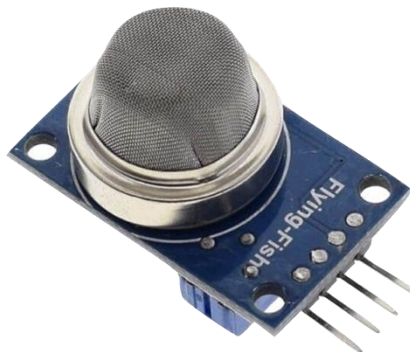
ตำแหน่งของการติดตั้งหัวพ่นน้ำจะทำการติดตั้งทั้งหมดจำนวน 6 หัวโดยจะทำการติดตั้งบริเวณโครงด้านบนของชั้นวางเห็ดชั้นละ 3 หัว ดังภาพที่ 3-15



ภาพที่ 3-16 ตำแหน่งของการติดตั้งตู้คอนโทรล

ตำแหน่งของการติดตั้งตู้คอนโทรลสำหรับจัดเก็บอุปกรณ์คอนโทรลเลอร์

3.4 อุปกรณ์ที่ใช้ในการจัดทำโครงการ



ภาพที่ 3-17 AIR QUALITY SENSOR MQ 135 เซนเซอร์ตรวจวัดคุณภาพอากาศ

คณะผู้จัดทำเลือกใช้เซนเซอร์ตรวจวัดคุณภาพอากาศ Air quality sensor MQ 135 มาใช้เป็นอุปกรณ์ที่ตรวจจับค่าก๊าซต่างๆภายในโรงเรือนเพาะเห็ดโดยจะทำการติดตั้งไว้บริเวณด้านบนของโรงเรือนเพาะเห็ด



ภาพที่ 3-18 เซนเซอร์ DHT 22 แสดงการใช้วัดค่าความชื้นและอุณหภูมิ

คณะผู้จัดทำเลือกใช้เซนเซอร์ DHT 11 มาเป็นอุปกรณ์ที่ใช้วัดค่าความชื้นและอุณหภูมิภายในโรงเรือนเพาะเห็ดเนื่องจาก DHT 11 มีโมดูลเซนเซอร์วัดความชื้นและอุณหภูมิภายในตัวเดียว



ภาพที่ 3-19 พัดลมระบายอากาศขนาด 8 นิ้ว

คณะผู้จัดทำเลือกใช้พัดลมระบายอากาศขนาด 8 นิ้ว มาติดตั้งภายในโรงเรือนเพาะเห็ดเพื่อใช้ระบาย อุณหภูมิที่สูงเกินค่ามาตรฐานที่การออกดอกของเห็ดนางฟ้า



ภาพที่ 3-20 มอเตอร์ปั๊ม 10.8บาร์



ภาพที่ 3-21 หัวพ่นหมอกทองเหลือง

คณะผู้จัดทำเลือกใช้มอเตอร์ปั๊ม 10.8 บาร์ ในการนำมาต่อเป็นอุปกรณ์ดูน้ำขึ้นมาให้รดน้ำผ่านหัวพ่นหมอกขนาด 0.3 มิลลิเมตร เพื่อเพิ่มความชื้นให้กับภายในโรงเรือนเพาะเห็ด

สรุปการทดสอบประสิทธิภาพการทำงานของระบบโรงเพาะเห็ดนางฟ้าภูฐาน

จากการทดสอบระบบเซนเซอร์ต่างๆภายในโรงเรือนเพาะเห็ดนางฟ้าภูฐานผู้จัดทำทำการทดสอบระบบเซนเซอร์ต่างๆทั้งหมดจำนวน 10 ครั้ง เพื่อตรวจสอบความเรียบร้อยในการทำงานของระบบเซนเซอร์ต่างๆภายในโรงเรือนซึ่งปรากฏว่าระบบเซนเซอร์ต่างๆภายในโรงเรือนสามารถทำงานได้สมบูรณ์

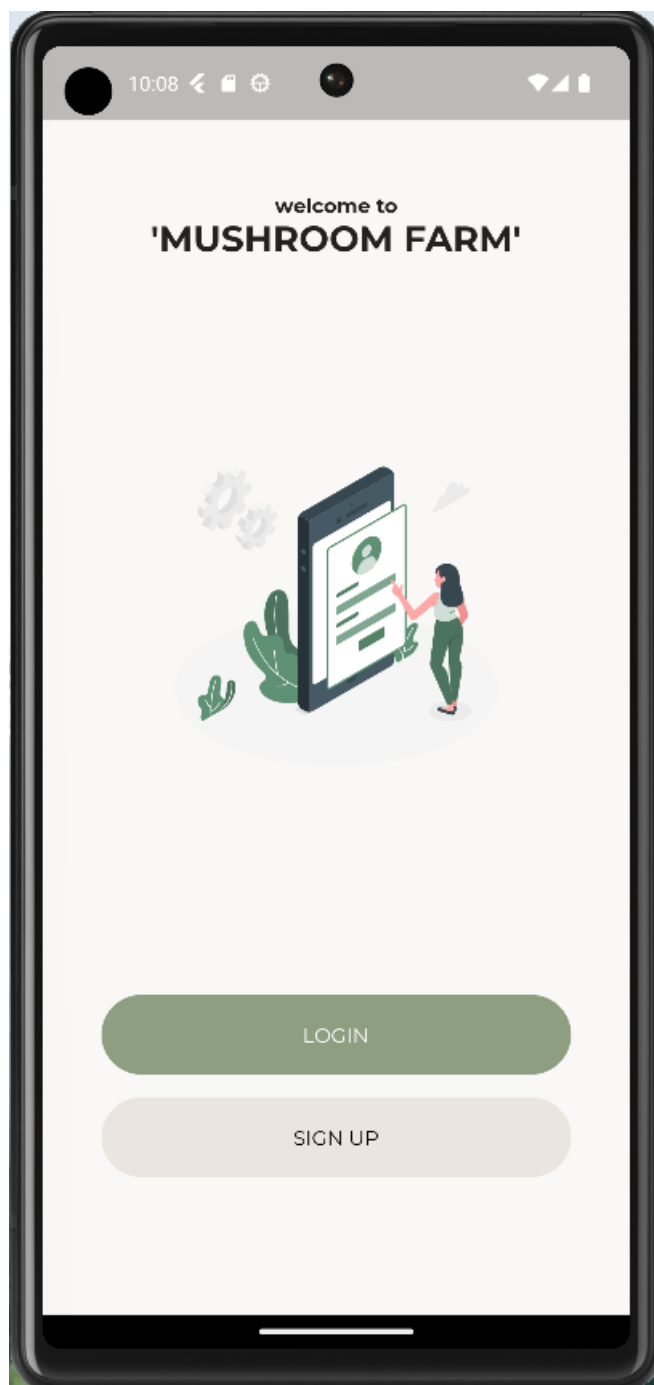


ภาพที่ 3-22 ทดสอบการทำงานของระบบหัวฉีดพ่นหมอก

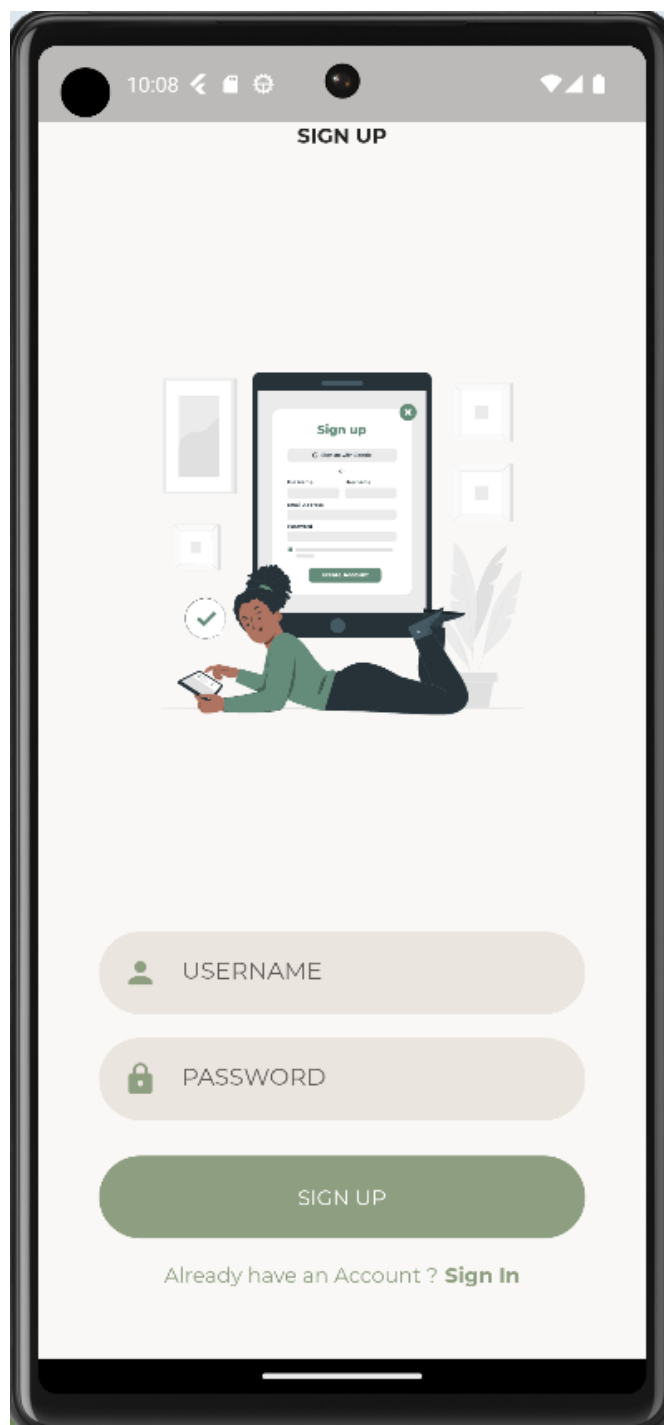
3.6 การทดสอบในส่วนของซอฟต์แวร์

ตารางที่ 3-3 ตารางการทดสอบในส่วนซอฟต์แวร์

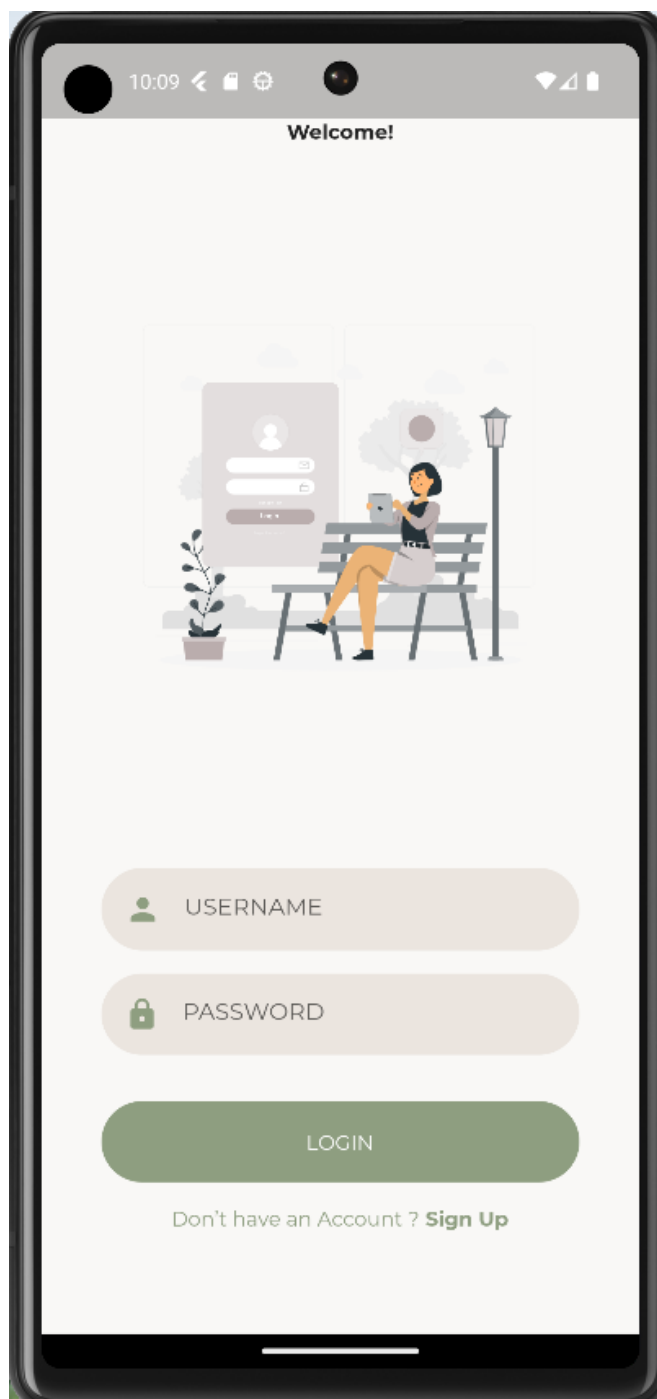
ฟังก์ชันของแอปพลิเคชัน	การใช้งาน
1.แอปพลิเคชันมีระบบสมาชิก	เข้าสู่ระบบได้ โดยบันทึกฐานข้อมูลลงใน Firebase เก็บข้อมูลเป็นอีเมลและพาสเวิร์ด ดังภาพที่ 3-24
2.แอปพลิเคชันสามารถดูค่าความชื้นและอุณหภูมิในโรงเพาะเห็ดได้	ดึงค่าความชื้นและอุณหภูมิจากเซนเซอร์ที่ถูกส่งค่าเข้า Firebase และนำมาแสดงบนแอปพลิเคชันดังภาพที่ 3-26
3.แอปพลิเคชันสามารถดูการเจริญเติบโตของเห็ดได้ผ่านกล้อง	ดังภาพที่ 3-26
4.แอปพลิเคชันสามารถดูค่าสภาพก๊าซในอากาศได้	ดึงค่าก๊าซจากเซนเซอร์ที่ถูกส่งค่าเข้า Firebase และนำมาแสดงบนแอปพลิเคชัน ดังภาพที่ 3-26
5.แอปพลิเคชันสามารถแจ้งเตือนได้	ดึงค่าก๊าซจากเซนเซอร์ที่ถูกส่งค่าเข้า Firebase และแจ้งเตือนผ่านแอปพลิเคชันเมื่อค่าเกินกว่าที่ควรจะเป็น ดังภาพที่ 3-27



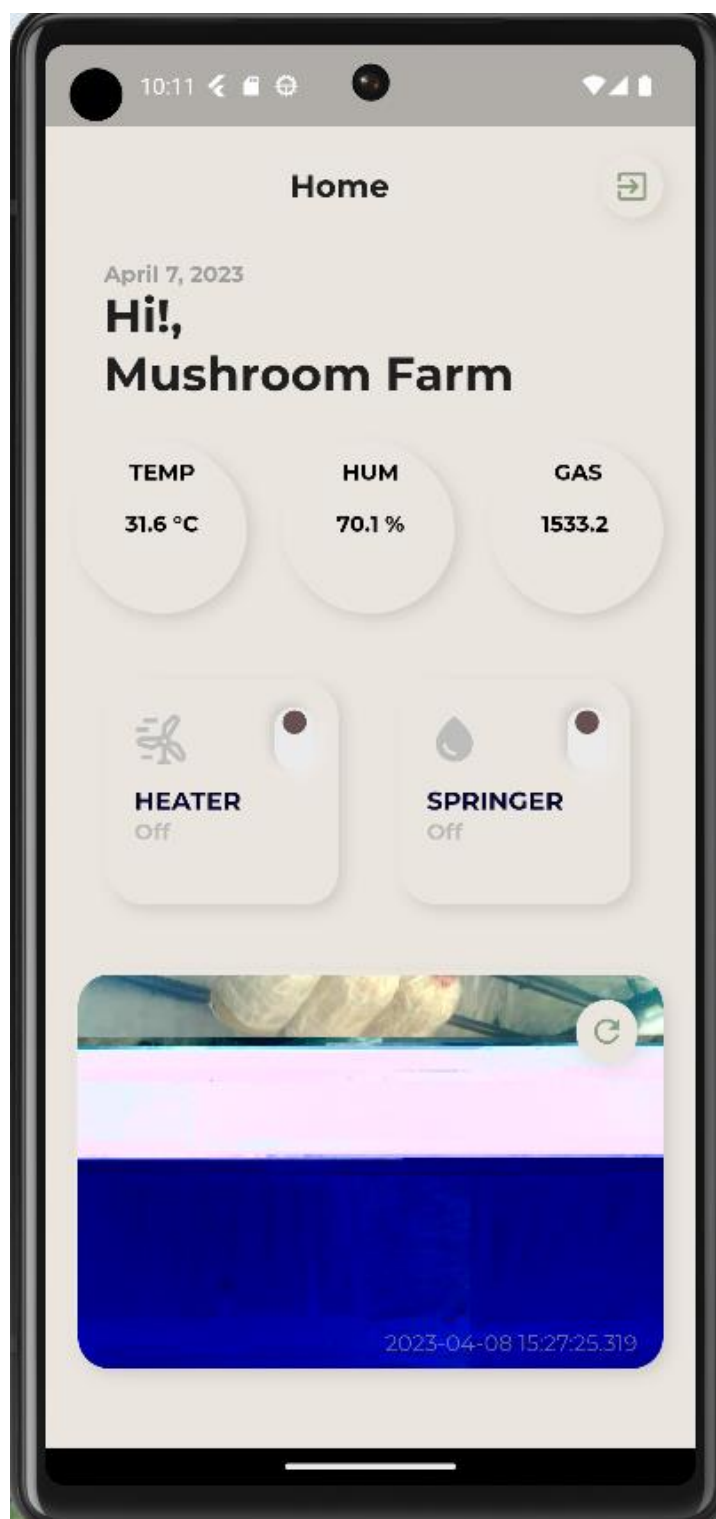
ภาพที่ 3-23 หน้าแรกที่เปิดขึ้นมาเพื่อเลือก LOG IN หรือ SIGN UP



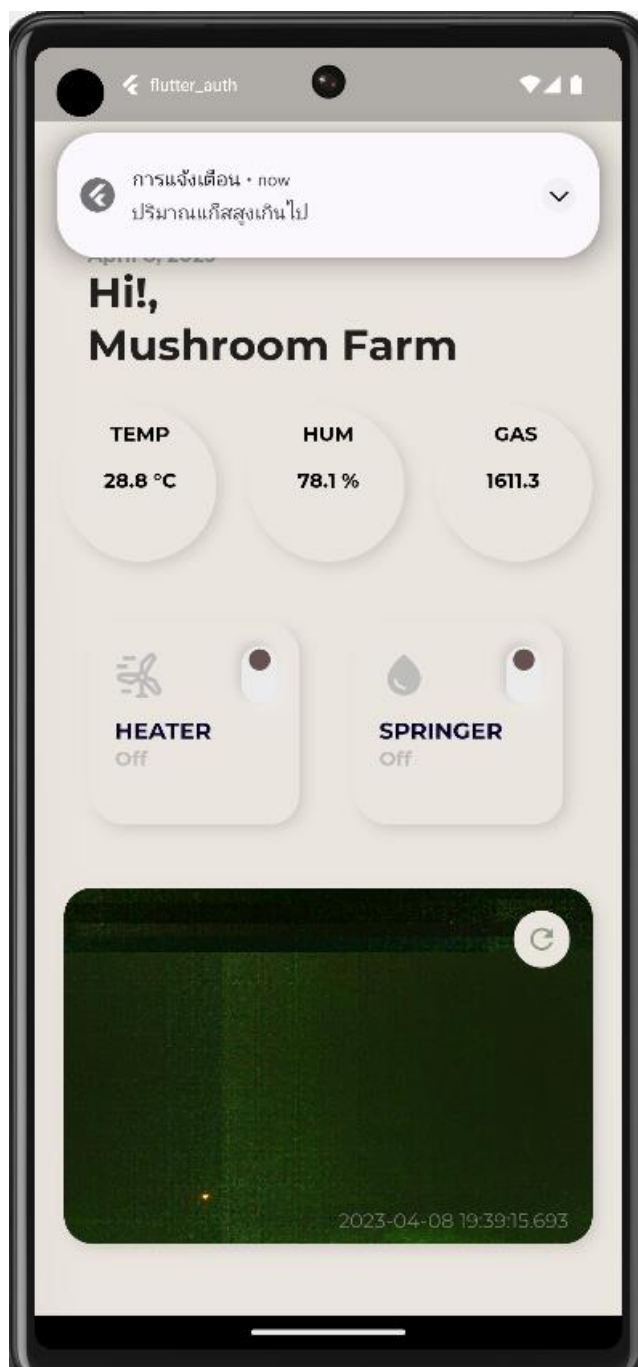
ภาพที่ 3-24 หน้าลงทะเบียน สำหรับผู้ที่ยังไม่เคยมีบัญชี



ภาพที่ 3-25 หน้าเข้าสู่ระบบ สำหรับผู้ที่มีบัญชีผู้ใช้แล้ว

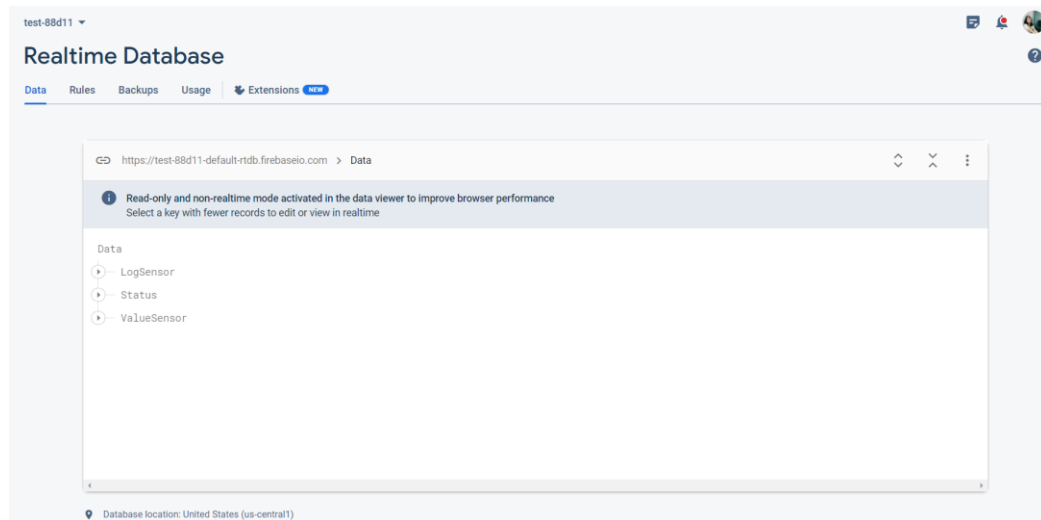


ภาพที่ 3-26 หน้าแสดงค่าต่างๆ ของระบบโรงเพาะเห็ด

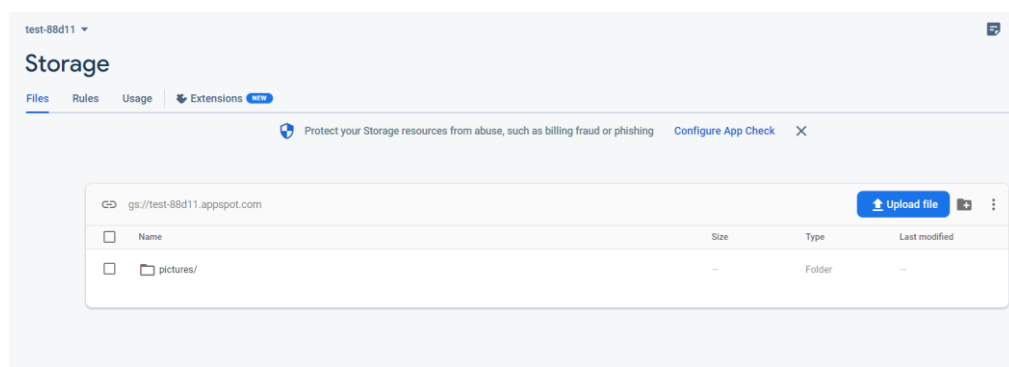


ภาพที่ 3-27 ระบบมีการแจ้งเตือนเมื่อค่าก๊าซเกิน

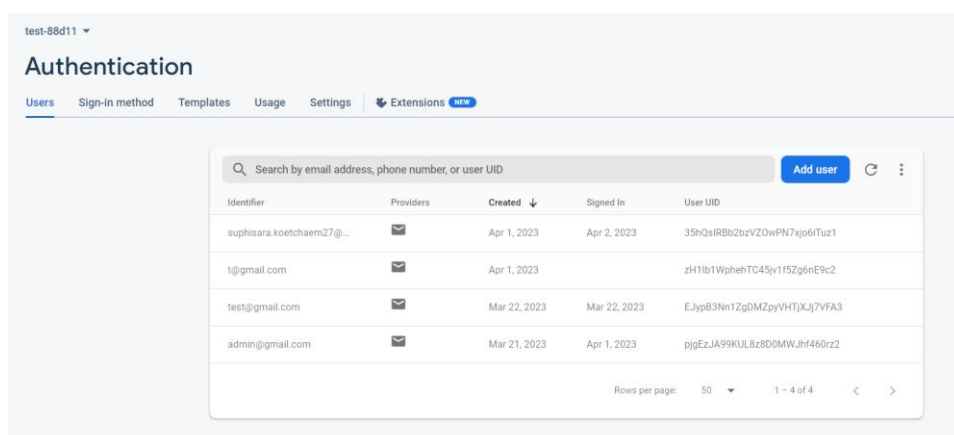
3.7 การเก็บฐานข้อมูล



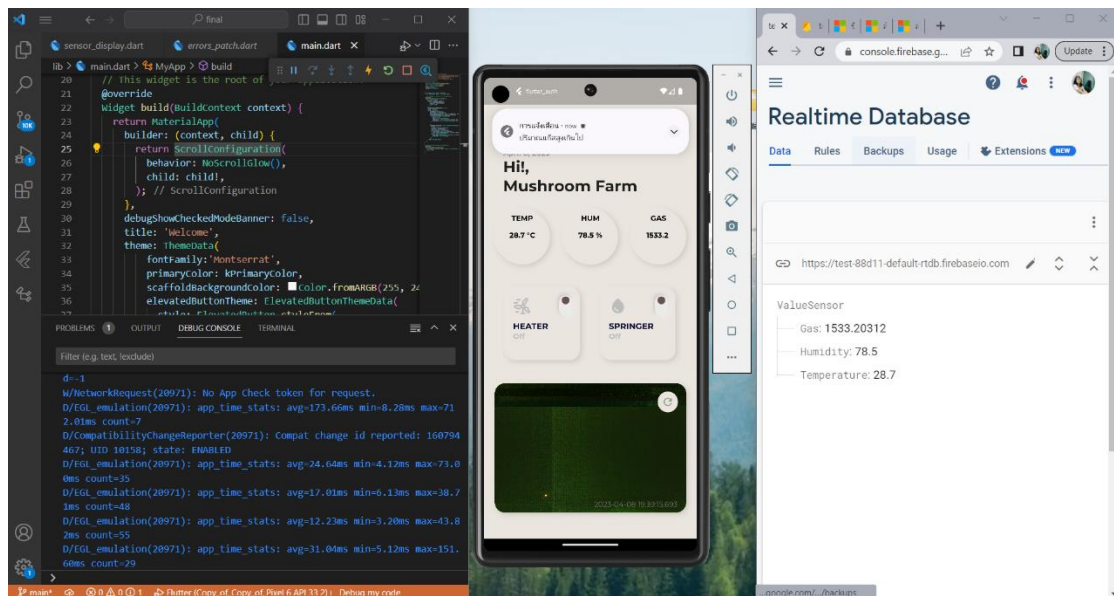
ภาพที่ 3-28 การเก็บ LOG ต่างๆของเซนเซอร์จะถูกเก็บอยู่ใน REALTIME DATABASE บน FIREBASE



ภาพที่ 3-29 การเก็บรูปภาพที่จะแสดงบนแอปพลิเคชันถูกเก็บอยู่ใน STORAGE บน FIREBASE



ภาพที่ 3-30 การเก็บข้อมูลของการเข้าสู่ระบบถูกเก็บอยู่ใน AUTHENTICATION บน FIREBASE



ภาพที่ 3-31 การยิงค่าขึ้น REALTIME DATANASE บน FIREBASE

บทที่ 4

ผลการทดลอง

คณะผู้จัดทำงานวิจัยจะทำการทำงานของโรงเพาะเห็ดนางฟ้าภูฏานระบบปิดกับการเพาะเห็ดนางฟ้าภูฏานระบบเปิดเพื่อเปรียบเทียบการออกดอกของเห็ดว่าระบบโรงเรือนที่คณะผู้จัดทำได้ทำขึ้นมานั้นมีการออกดอกของเห็ดที่ดีกว่าหรือไม่

4.1 ผลการทดลองการเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฏานระยะเวลา 3 วัน

ตารางที่ 4-1 ผลการทดลองการเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฏาน 3 วัน

เพาะเห็ดนางฟ้าภูฏานระบบปิด	ภายในก้อนเห็ดที่นำมาเพาะเริ่มการเกิดเชื้อขึ้น
เพาะเห็ดนางฟ้าภูฏานระบบเปิด	เห็ดยังไม่ออกดอก



ภาพที่ 4-1 การเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฏาน 3 วัน หลังเพาะ

4.2 ผลการทดลองการเปลี่ยนแปลงของก้านเห็ดนางฟ้าภูฐานระยะเวลา 5 วัน

ตารางที่ 4-2 ผลการทดลองการเปลี่ยนแปลงของก้านเห็ดนางฟ้าภูฐาน ระยะเวลา 5 วัน

เพาะเห็ดนางฟ้าภูฐานระบบปิด	เห็ดเริ่มออกดอกมาแต่ยังไม่สมบูรณ์เต็มที่
เพาะเห็ดนางฟ้าภูฐานระบบเปิด	เห็ดยังไม่ออกดอก



ภาพที่ 4-2 การเปลี่ยนแปลงของก้านเห็ดนางฟ้าภูฐาน 5 วัน หลังเพาะ

4.3 ผลการทดลองการเปลี่ยนแปลงของก้านเห็ดนางฟ้าภูฐาน 7 วัน หลังเพาะ

ตารางที่ 4-3 ผลการทดลองการเปลี่ยนแปลงของก้านเห็ดนางฟ้าภูฐาน ระยะเวลา 7 วัน

การเพาะเห็ดนางฟ้าภูฐานระบบปิด	ปริมาณของเห็ดที่ออกดอกมาจำนวนเยอะขึ้นและมีบางข้อที่สมบูรณ์พร้อมเก็บ
การเพาะเห็ดนางฟ้าภูฐานระบบเปิด	เห็ดยังไม่ออกดอก



ภาพที่ 4-3 การเปลี่ยนแปลงของก้อนเห็ดนางฟ้าภูฐาน 7 วัน หลังเพาะ

4.4 ผลการเกิดโรคเชื้อราเขียว

ตารางที่ 4-4 บันทึกผลการเกิดโรคเชื้อราเขียว

เพาะเห็ดนางฟ้าภูฐานระบบปิด	ไม่มีการเกิดโรคเชื้อราเขียวในก้อนเห็ดนางฟ้าภูฐานที่เพาะในระบบปิด
เพาะเห็ดนางฟ้าภูฐานระบบเปิด	มีการเกิดโรคเชื้อราเขียวในก้อนเห็ดนางฟ้าภูฐานที่เพาะในระบบเปิด

4.5 ผลการเสียหายของก้อนเห็ดนางฟ้าภูฐานที่เกิดจากแมลง

ตารางที่ 4-5 บันทึกผลการเสียหายของก้อนเห็ดนางฟ้าภูฐานที่เกิดจากแมลง

เพาะเห็ดนางฟ้าภูฐานระบบปิด	ไม่มีการเสียหายของก้อนเห็ดนางฟ้าภูฐานที่เพาะในระบบปิด
เพาะเห็ดนางฟ้าภูฐานระบบเปิด	ไม่มีการเสียหายของก้อนเห็ดนางฟ้าภูฐานที่เพาะในระบบเปิด

4.6 คุณภาพของเห็ดนางฟ้าภูฐาน

ตารางที่ 4-6 บันทึกผลคุณภาพของเห็ดนางฟ้าภูฐาน

เพาะเห็ดนางฟ้าภูฐานระบบปิด	เห็ดนางฟ้าภูฐานที่ออกดอกมามีสีน้ำตาล หมายความว่าเห็ดมีคุณภาพที่ดี
เพาะเห็ดนางฟ้าภูฐานระบบเปิด	เห็ดยังไม่ออกดอก

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุปและวิจารณ์ผลการทดลอง

ผู้จัดทำได้มีการสร้างโรงเรือนเพาะเห็ดนางฟ้าภูฐานโดยมีการนำเอาเทคโนโลยี IoT เข้ามาประยุกต์ใช้ในการควบคุมปัจจัยต่างๆ ที่มีผลต่อการออกดอกของเห็ดนางฟ้าภูฐาน โรงเรือนที่จัดทำขึ้นได้ทดสอบแล้วว่าสามารถเพาะเห็ดนางฟ้าภูฐานได้จริงเนื่องจากโรงเรือนที่จัดทำขึ้นสามารถควบคุมปัจจัยที่มีผลต่อการออกดอกของเห็ดนางฟ้าภูฐานได้ไม่ว่าจะเป็นในส่วนของการควบคุมความชื้นให้มีค่าความชื้นสัมพัทธ์ระหว่าง 70% ถึง 90% มีอุณหภูมิระหว่าง 27 °C ถึง 30 °C และอากาศที่ถ่ายเทตลอดได้แบบอัตโนมัติและผู้ใช้งานยังสามารถควบคุมระบบการทำงานภายในโรงเรือนเพาะเห็ดนางฟ้าภูฐานได้ผ่านอุปกรณ์พกพา เช่น การเปิด-ปิดระบบน้ำ การเปิด-ปิดพัดลมระบายอากาศ การเปิด-ปิดฮีตเตอร์ และการดูการออกดอกของเห็ดได้ ซึ่งสามารถสั่งการได้จากแอปพลิเคชันบนสมาร์ตโฟนหรืออุปกรณ์พกพาได้ จากการทดสอบข้างต้นทำให้เห็นว่าโรงเรือนเพาะเห็ดนางฟ้าภูฐานที่ผู้จัดทำให้จัดทำขึ้นนั้นเป็นไปตามวัตถุประสงค์ในข้อที่ 1.2.1 และข้อที่ 1.2.2

ผลการทดลองจากตารางที่ 3-2 แบบทดสอบประสิทธิภาพการทำงานของระบบโรงเรือนเพาะเห็ดนางฟ้าภูฐานจะเป็นการทดสอบอุปกรณ์เซนเซอร์ ทดสอบการทำงานของระบบหัวฉีดพ่นหมอกที่ติดตั้งภายในโรงเรือนเพาะเห็ดนางฟ้าภูฐานและทดสอบการส่งค่าที่ได้จากอุปกรณ์เซนเซอร์ไปเก็บไว้ยังฐานข้อมูลจากการทดสอบระบบอุปกรณ์เซนเซอร์ ระบบหัวฉีดพ่นหมอกที่ทำการติดตั้งไปนั้นสามารถทำงานได้ตรงตามเป้าหมายที่คาดหวังไว้และในส่วนของการทดสอบการส่งค่าที่ได้จากอุปกรณ์เซนเซอร์ไปเก็บไว้ในฐานข้อมูลก็สามารถส่งค่าไปเก็บไว้ยัง Firebase ได้ตามเป้าหมายที่คาดหวังไว้

ผลการทดลองจากตารางที่ 3-3 ฟังก์ชันของแอปพลิเคชันตามขอบเขตที่ได้กำหนดไว้ มีการสมัครสมาชิกใช้การลงทะเบียนด้วยอีเมลและรหัสผ่าน 6 ตัวขึ้นไปและเก็บค่าไว้ในฐานข้อมูล สามารถดูค่าอุณหภูมิ, ความชื้น, ก๊าซ และ กล้องได้ผ่านแอปพลิเคชันโดยดึงค่าจากฐานข้อมูลที่ฝั่งเซนเซอร์ส่งมา และสามารถแจ้งเตือนได้เมื่อค่าก๊าซเกิน การแสดงค่า การแจ้งเตือน รวมถึงการสมัครสมาชิกและเข้าสู่ระบบเป็นไปตามเป้าหมายที่ผู้จัดทำได้คาดหวังไว้

ผลการทดลองจากตารางที่ 4-1 ตารางที่ 4-2 และตารางที่ 4-3 จะเป็นการติดตามดูผลการออกดอกของเห็ดนางฟ้าภูฐาน ระยะเวลา 3 วัน 5 วัน และ 7 วัน ตามลำดับโดยจะเป็นการนำเอาเห็ดนางฟ้าภูฐานมาเพาะภายในระบบปิดของโรงเรือนที่จัดทำขึ้นและเพาะในระบบเปิด เพื่อเป็นการเปรียบเทียบให้เห็นได้อย่างชัดเจนยิ่งขึ้นว่าโรงเรือนเพาะเห็ดนางฟ้าภูฐานที่จัดทำขึ้นนั้นมีประสิทธิภาพหรือไม่จากผลการทดลองนั้นจะเห็นได้ว่าการเพาะเห็ดนางฟ้าภูฐานในระบบปิดของโรงเรือนและระบบเปิดมีผลการทดลองที่ต่างกันอย่างชัดเจน การเพาะเห็ดนางฟ้าภูฐานในระบบปิดของโรงเรือนก่อนเชื้อของเห็ดมีการเปลี่ยนแปลงตั้งแต่ 3 วันแรกและมีการออกดอกของเห็ดในระยะเวลาต่อมาแต่ในการเพาะเห็ดนางฟ้าภูฐานในระบบเปิดนั้นไม่มีการออกดอกของเห็ด จากผลการทดลองนี้ผู้จัดทำสรุปได้ว่าโรงเรือนเพาะเห็ดนางฟ้าภูฐานที่ได้จัดทำขึ้นนั้นมีประสิทธิภาพในการควบคุมปัจจัยในการออกดอกของเห็ดได้จริง

ผลการทดลองจากตารางที่ 4-4 จะเป็นการติดตามดูผลการเกิดโรคเชื้อราเขียวในก้อนเห็ดนางฟ้าภูฐานที่นำมาเพาะภายในระบบปิดของโรงเรือนที่จัดทำขึ้นและเพาะในระบบเปิดจากผลการทดลองจะเห็นได้ว่าก้อนเห็ดนางฟ้าภูฐานที่เพาะในระบบปิดนั้นจะไม่มีการเกิดโรคเชื้อราเขียวเนื่องจากสามารถควบคุมปัจจัยที่ทำให้เกิดโรคเชื้อราเขียวภายในโรงเรือนได้แต่ก้อนเห็ดนางฟ้าภูฐานที่เพาะในระบบเปิดนั้นจะมีการเกิดเชื้อราเกิดขึ้นเนื่องจากไม่สามารถควบคุมปัจจัยที่ทำให้เกิดโรคเชื้อราเขียวได้จากผลการทดลองนี้ผู้จัดทำสรุปได้ว่าโรงเรือนเพาะเห็ดนางฟ้าภูฐานที่ได้จัดทำขึ้นนั้นมีประสิทธิภาพในการเพาะเห็ดนางฟ้าภูฐานเนื่องจากสามารถลดการเกิดโรคเชื้อราเขียวที่ส่งผลทำให้ก้อนเห็ดเกิดการเสียหายได้

ผลการทดลองจากตารางที่ 4-5 จะเป็นการติดตามดูผลการเกิดความเสียหายของก้อนเห็ดนางฟ้าภูฐานที่เกิดจากแมลงโดยจะติดตามผลจากก้อนเห็ดนางฟ้าภูฐานที่นำมาเพาะภายในระบบปิดของโรงเรือนที่จัดทำขึ้นและเพาะในระบบเปิดจากผลการทดลองจะเห็นได้ว่าก้อนเห็ดนางฟ้าภูฐานที่เพาะในระบบปิดนั้นก้อนเห็ดนางฟ้าภูฐานไม่เกิดการเสียหายจากแมลงเนื่องจากมีการจัดโรงเรือนที่ปิดมิดชิดแต่ในส่วนของก้อนเห็ดนางฟ้าภูฐานที่เพาะในระบบเปิดเกิดจากเสียหายจากแมลงบางก้อนเนื่องจากสถานที่ในการเพาะนำก้อนเห็ดนางฟ้าภูฐานไปเพาะนั้นมีการปิดที่ไม่มิดชิด จากผลการทดลองนี้ผู้จัดทำสรุปได้ว่าโรงเรือนเพาะเห็ดนางฟ้าภูฐานที่ได้จัดทำขึ้นนั้นมีประสิทธิภาพในการเพาะเห็ดนางฟ้าภูฐาน เนื่องจากสามารถลดการเกิดความเสียหายของก้อนที่จากแมลงได้

ผลการทดลองจากตารางที่ 4-6 จะเป็นการติดตามคุณภาพของเห็ดนางฟ้าภูฐาน โดยจะติดตามผลจากก้อนเห็ดนางฟ้าภูฐานที่นำมาเพาะภายในระบบปิดของโรงเรือนที่จัดทำขึ้นและเพาะในระบบเปิดจากผลการทดลองจะเห็นว่าก้อนเห็ดนางฟ้าภูฐานที่เพาะในระบบปิดนั้นดอกของเห็ดนางฟ้ามีลักษณะสีเทาน้ำตาล ขนาดใหญ่ อวบ ไม่ลีบแบน ซึ่งจะบ่งบอกถึงว่าดอกเห็ดนั้นมีลักษณะที่สมบูรณ์คุณภาพดีจากผลการทดลองนี้ผู้จัดทำสรุปได้ว่าโรงเรือนเพาะเห็ดนางฟ้าภูฐานที่ได้จัดทำขึ้นนั้นมีประสิทธิภาพในการเพาะเห็ดนางฟ้าภูฐานเนื่องจากลักษณะของดอกเห็ดที่ออกมาสมบูรณ์ มีคุณภาพที่ดี ซึ่งสามารถบ่งบอกได้ว่าโรงเรือนเพาะเห็ดนางฟ้าภูฐานนั้นสามารถควบคุมปัจจัยในการออกดอกของเห็ดนางฟ้าภูฐานได้ดี

5.2 ข้อเสนอแนะและแนวทางในการพัฒนา

- 5.2.1 ในส่วนของอุปกรณ์เซนเซอร์ต่างๆควรใช้อุปกรณ์ที่มีความทนทาน
- 5.2.2 พื้นที่ในการจัดวางโรงเรือนเพาะเห็ดนางฟ้าภูฐานควรเป็นที่ไม่โล่งแจ้งและพื้นควรจะสามารถกักเก็บน้ำได้เพื่อให้ภายในโรงเรือนมีการกักเก็บความชื้นที่ดี
- 5.2.3 การติดตั้งหัวพ่นหมอกควรจะมีการติดตั้งในองศาที่ละอองน้ำไม่พ่นเข้าปากก้อนเชื้อเห็ด
- 5.2.4 ในส่วนของซอฟต์แวร์การใช้ flutter มีปัญหาเกิดขึ้นบ่อยเรื่องเวอร์ชันไม่ตรงกับที่ระบบต้องการ อาจทำให้เสียเวลาไปเยอะกับตรงนี้ แต่ถ้าทำความเข้าใจและแก้ปัญหาได้ก็เป็นซอฟต์แวร์ที่ใช้เขียน mobile application ได้ดี
- 5.2.5 การนำข้อมูลจาก Database ขึ้นมาหน้า Application บางครั้งอาจแสดงค่าที่ไม่สมบูรณ์ ดังนั้นจึงต้องสร้างเงื่อนไขเพิ่มเติมเพื่อแก้ไขปัญหาดังกล่าว

บรรณานุกรม

- [1] กระทรวงเกษตรและสหกรณ์. (2564). รายงานสรุปข้อมูลสำคัญของกระทรวงเกษตรและสหกรณ์. <https://data.moac.go.th/>
- [2] ไทยรัฐออนไลน์. (2564). สู้ช็อก 3 ศพ ดับคาโรงเพาะเห็ด คาดปล่อยก๊าซซอบเชื้อ ทำชาดออกซิเจน. <https://www.thairath.co.th/news/local/northeast/2137742>
- [3] นายศิวกร จินดารัตน์. (2557). ระบบจัดการฟาร์มไก่อัจฉริยะด้วยระบบเออาร์พีและอาอคูนโน. กรุงเทพมหานคร:มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
- [4] บมจ.ธนาคารกรุงเทพ.(2562). 5 กรณีการใช้ IoT เพื่อการทำเกษตรอัจฉริยะ. <https://www.bangkokbanksme.com/en/lot-smart-farm>
- [5] ฟาร์มเห็ดกลางกรุง. (2564). เพาะเห็ดขายอาชีพเสริมรายได้ดี. <https://r-sim100.com/11926>
- [6] วีรศักดิ์ ฟองเงิน, สุรพงษ์ เพ็ชรหาญ, และรัฐสิทธิ์ ยะจ่อ. (2561). การประยุกต์ใช้เทคโนโลยีไอโอทีควบคุมฟาร์มอัจฉริยะในโรงเรือนเพาะเห็ดนางฟ้า. วารสารวิชาการการจัดการเทคโนโลยีสารสนเทศและนวัตกรรม คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏมหาสารคาม, ฉบับที่ 5 , 173.
- [7] ไอแซก. DHT22 เซนเซอร์อุณหภูมิและความชื้นที่แม่นยำ. <https://www.hwlibre.com/th/dht22/>
- [8] Cibertice. เซนเซอร์วัดความสว่างความเข้มแสง LDR Photoresistor Sensor Module. <https://www.cybertice.com/product>
- [9] iMacbaszii's Blog (2014). รู้จักกับ Swift ภาษาที่ทำให้การเขียนแอปบน iOS, OS X ง่ายขึ้น. <https://www.macthai.com/2014/06/07/introduction-to-swift-programming-language-from-apple/>

[10] Mushroom2010. เห็นนางฟ้าภูฐาน. <https://mushrooms2010.wordpress.com>

[11] Yongyut Srisuban. (2562). Embedded System.

<https://medium.com/@yongyutsrisuban/embedded-system-267ceee1665e>

ภาคผนวก ก

การเขียน source code ของระบบการทำงานภายในโรงเรียนเพาะเห็ดนางฟ้าภูฐาน

เนื้อหาภาคผนวก ก

```
//Sensor
#include "DHT.h"
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>

#define DHTPIN D2

#define DHTTYPE DHT22

int m = 0;
DHT dht(DHTPIN, DHTTYPE);

//Gas//
#define MQ135_PIN A0
int CO_pin = A0; // analog input pin connected to the CO sensor
float Vout, ppm;

//Fan//
int A = D7 ;
int B = D8 ;

//Pum & Heater//
int pum = D5 ;
int hea = D6 ;

// Replace with your network credentials
const char* ssid = "TP-Link_7EE6";
const char* password = "11752073";
```

```

// Replace with your Firebase project's URL and secret key
#define FIREBASE_HOST "test-88d11-default-rtdb.firebaseio.com/0"
#define FIREBASE_AUTH "v1kq69h2B7AeN2N5Ta1Giv2lsfBAAIC0FiwatWUI"

// Initialize FirebaseESP8266 object
FirebaseData firebaseData;

// Define the interval to send data to Firebase (in milliseconds)
const unsigned long INTERVAL = 3600000; // 1 Hour
//const unsigned long INTERVAL = 600000; // 10 Mins
//const unsigned long INTERVAL = 60000; // 1 Mins

// Define the last time the data was sent
unsigned long lastTime = 0;

void setup() {

// Connect to Wi-Fi network
WiFi.begin(ssid, password);
Serial.print("Connecting to Wi-Fi");
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println();
Serial.print("Connected to Wi-Fi, IP address: ");
Serial.println(WiFi.localIP());

// Connect to Firebase
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Serial.println("Connected to Firebase");

```

```
pinMode (A,OUTPUT);
pinMode (B,OUTPUT);

pinMode (pum,OUTPUT);
pinMode (hea,OUTPUT);

Serial.begin(115200);

dht.begin();
}

void loop() {

float humidity = dht.readHumidity();
float temperature = dht.readTemperature();

Serial.print(humidity);
Serial.println("%");
Serial.print(temperature);
Serial.println("°C");

//Gas//
Vout = analogRead(CO_pin); // read the sensor output voltage
ppm = (Vout / 1024.0) * 10000.0; // convert voltage to ppm concentration using
sensor calibration curve
Serial.print("CO2: ");
Serial.print(ppm);
Serial.println(" ppm");
```



```

float ts = (temperature);
int hs = (humidity);

unsigned long currentTime = millis();
if (currentTime - lastTime >= INTERVAL) {
    lastTime = currentTime;

    // Send data to Firebase >> push //

    if (Firebase.pushFloat(firebaseData, "/Data/LogSensor/Temperature", temperature)){
        Serial.println("Temperature Data sent to Firebase");
    }
    else {
        digitalWrite(pum, LOW);
        (Firebase.pushString(firebaseData, "/Data/LogSensor/Temperature", "NaN") );
        Serial.println("Error sending data to Firebase");
        Serial.println(firebaseData.errorReason());
    }

    if (Firebase.pushFloat(firebaseData, "/Data/LogSensor/Humidity", humidity)) {
        Serial.println("Humidity Data sent to Firebase");
    }
    else {
        digitalWrite(pum, LOW);
        (Firebase.pushString(firebaseData, "/Data/LogSensor/Humidity", "NaN") );
        Serial.println("Error sending data to Firebase");
        Serial.println(firebaseData.errorReason());
    }

    if (Firebase.pushInt(firebaseData, "/Data/LogSensor/Gas(Co2)", ppm )) {
        Serial.println("Gas(Co2) sent to Firebase");
    }

```

```

}
else {
    digitalWrite(pum, LOW);
    (Firebase.pushString(firebaseData, "/Data/LogSensor/Gas(Co2)", "NaN") );
    Serial.println("Error sending data to Firebase");
    Serial.println(firebaseData.errorReason());
}

    Serial.println("Data sent to Firebase");
}

if(ts >= 30 )
{
    analogWrite(A, 255); //Fan//
    analogWrite(B, 255);
    Serial.println("Fan Lavel 2 : On...");
    Firebase.setString(firebaseData, "/Data/Status/FanStatus", "Fan Lavel 2 : On");
}

if(ts >= 24 && ts < 30)
{
    analogWrite(A, 200);
    analogWrite(B, 200);
    Serial.println("Fan Lavel 1 : On...");
    Firebase.setString(firebaseData, "/Data/Status/FanStatus", "Fan Lavel 1 : On");
}

if(ts <= 23)
{
    digitalWrite(hea, HIGH); //Heater
    Serial.println("Heater On...");
}

```

```

    Firebase.setString(firebaseData, "/Data/Status/HeaterStatus", "Heater On");
}
if(ts >= 24)
{
    digitalWrite(hea, LOW); //Heater
    Serial.println("Heater Off...");
    Firebase.setString(firebaseData, "/Data/Status/HeaterStatus", "Heater Off");
}

if(hs >= 70 && hs <= 90)
{
    digitalWrite(pum, LOW); //pum
    Serial.println("Pump Off...");
    Firebase.setString(firebaseData, "/Data/Status/WaterPumpStatus", "Pump Off");
}

if(hs <=69 )
{
    digitalWrite(pum, HIGH); //pum
    Serial.println("Pump Off...");
    Firebase.setString(firebaseData, "/Data/Status/WaterPumpStatus", "Pump On");
}

/* if(ts >= 32 )
{
    digitalWrite(pum, HIGH); //Pum
    Serial.println("Pump On");
    Firebase.setString(firebaseData, "/Data/Status/WaterPumpStatus01", "The pump
works because of the high temperature.");

```

```

    delay(10000);
}
else{
    digitalWrite(pum, LOW); //Pum
    Serial.println("Pump Off");
    Firebase.setString(firebaseData, "/Data/Status/WaterPumpStatus01", "Pump Off");
}*/
// Send data to Firebase >> set //

if (Firebase.setFloat(firebaseData, "/Data/ValueSensor/Temperature", temperature))
{
    Serial.println("ValueTemperature Data sent to Firebase");
}
else { digitalWrite(pum, LOW);
    (Firebase.setString(firebaseData, "/Data/ValueSensor/Temperature", "NaN") );
    Serial.println("Error sending data to Firebase Value");
    Serial.println(firebaseData.errorReason());
}

if (Firebase.setFloat(firebaseData, "/Data/ValueSensor/Humidity", humidity)) {
    Serial.println("ValueHumidity Data sent to Firebase");
}
else {digitalWrite(pum, LOW);
    (Firebase.setString(firebaseData, "/Data/ValueSensor/Humidity", "NaN") );
    Serial.println("Error sending data to Firebase Value");
    Serial.println(firebaseData.errorReason());
}

if (Firebase.setFloat(firebaseData, "/Data/ValueSensor/Gas", ppm)) {
    Serial.println("ValueHumidity Data sent to Firebase");
}

```

```
else {digitalWrite(pum, LOW);  
      (Firebase.setString(firebaseData, "/Data/ValueSensor/Gas", "NaN" ) );  
      Serial.println("Error sending data to Firebase Value");  
      Serial.println(firebaseData.errorReason());  
    }  
  }
```

ภาคผนวก ข

การเขียน source code ของแอปพลิเคชันควบคุมโรงเรือนเพาะเห็ดนางฟ้าภูฐาน

เนื้อหาภาคผนวก ข

```

already_have_an_account_acheck.dart X
lib > components > already_have_an_account_acheck.dart > ...
1 import 'package:flutter/material.dart';
2 import 'package:flutter_auth/constants.dart';
3
4 class AlreadyHaveAnAccountCheck extends StatelessWidget {
5   final bool login;
6   final Function? press;
7   const AlreadyHaveAnAccountCheck({
8     Key? key,
9     this.login = true,
10    required this.press,
11  }) : super(key: key);
12
13  @override
14  Widget build(BuildContext context) {
15    return Row(
16      mainAxisAlignment: MainAxisAlignment.center,
17      children: <Widget>[
18        Text(
19          login ? "Don't have an Account ? " : "Already have an Account ? ",
20          style: const TextStyle(color: kPrimaryColor),
21        ), // Text
22        GestureDetector(
23          onTap: press as void Function()?,
24          child: Text(
25            login ? "Sign Up" : "Sign In",
26            style: const TextStyle(
27              color: kPrimaryColor,
28              fontWeight: FontWeight.bold,
29            ), // TextStyle
30          ), // Text
31        ), // GestureDetector
32      ], // <Widget>[]
33    ); // Row
34  }
35

```

```

background.dart X
lib > components > background.dart > Background > build
1 import 'package:flutter/material.dart';
2
3 class Background extends StatelessWidget {
4   final Widget child;
5   const Background({
6     Key? key,
7     required this.child,
8     this.topImage = "assets/images/main_top.png",
9     this.bottomImage = "assets/images/login_bottom.png",
10  }) : super(key: key);
11
12   final String topImage, bottomImage;
13
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       //resizeToAvoidBottomInset: false,
18       body: Container(
19         width: double.infinity,
20         height: MediaQuery.of(context).size.height,
21         child: SafeArea(child: child),
22       ), // Container
23     ); // Scaffold
24   }
25 }
26

```

```

notification.dart •
lib > components > notification.dart > NotificationService > onDisplay
1  import 'package:awesome_notifications/awesome_notifications.dart';
2  import 'package:flutter/material.dart';
3
4  class NotificationService {
5    static Future<void> init() async {
6      await AwesomeNotifications().initialize(
7        null,
8        [
9          NotificationChannel(
10           channelGroupKey: "high_importance_channel",
11           channelKey: "high_importance_channel",
12           channelName: "notification",
13           channelDescription: "notiication",
14           defaultColor: Colors.white,
15           ledColor: Colors.white,
16           importance: NotificationImportance.Max,
17           channelShowBadge: false,
18           onlyAlertOnce: false,
19           playSound: true,
20           criticalAlerts: true
21         )
22       ],
23       channelGroups: [
24         NotificationChannelGroup(
25           channelGroupKey: "high_importance_channel",
26           channelGroupName: "group"
27         )
28       ],
29     );
30     await AwesomeNotifications().isNotificationAllowed().then((isAllow) {
31       if (!isAllow) {
32         AwesomeNotifications().requestPermissionToSendNotifications();
33       }
34     });
35     await AwesomeNotifications().setListeners(
36       onActionReceivedMethod: onAction,
37       onDismissActionReceivedMethod: onDismiss,
38       onNotificationCreatedMethod: onCreate,
39       onNotificationDisplayedMethod: onDisplay,
40     );
41   }
42   static Future<void> onAction(re) async {
43   }
44   static Future<void> onDismiss(re) async {
45   }
46   static Future<void> onCreate(re) async {
47   }
48   static Future<void> onDisplay(re) async {
49   }
50   static showNotification({title = "", body = ""}) async {
51     await AwesomeNotifications().createNotification(
52       content: NotificationContent(
53         id: -1,
54         channelKey: "high_importance_channel",
55         title: title,
56         body: body
57       )
58     );
59   }
60 }

```



```

notification.dart • control_button.dart •
lib > LandingScreen > components > control_button.dart > ControlButton
1  import 'package:flutter/material.dart';
2  import 'package:flutter_auth/constants_2.dart';
3
4  class ControlButton extends StatefulWidget {
5    const ControlButton({
6      required Key key,
7      required this.size,
8      required this.icon,
9      required this.title,
10   }) : super(key: key);
11    final Size size;
12    final IconData icon;
13    final String title;
14    @override
15    _ControlButtonState createState() => _ControlButtonState();
16  }
17  class _ControlButtonState extends State<ControlButton> {
18    bool isSelected = false;
19    @override
20    Widget build(BuildContext context) {
21      return Column(
22        children: [
23          GestureDetector(
24            onTap: () {
25              setState(() {
26                isSelected = !isSelected;
27              });
28            },
29            child: Container(
30              height: widget.size.height * 0.105,
31              width: widget.size.width * 0.21,
32              decoration: BoxDecoration(
33                color: isSelected ? kOrangeColor : Colors.white,
34                borderRadius: BorderRadius.circular(20),
35                boxShadow: [
36                  BoxShadow(
37                    color: isSelected

```

```

38                    ? kOrangeColor.withOpacity(0.5)
39                    : Colors.grey.shade200,
40                    blurRadius: 30,
41                    offset: Offset(5, 5),
42                  ), // BoxShadow
43                ],
44              ), // BoxDecoration
45              child: Icon(
46                widget.icon,
47                color:
48                  isSelected ? Colors.white : kDarkGreyColor.withOpacity(0.6),
49                size: 45,
50              ), // Icon
51            ), // Container
52          ), // GestureDetector
53          SizedBox(height: widget.size.height * 0.005),
54          Text(
55            widget.title,
56            textAlign: TextAlign.center,
57            style: TextStyle(
58              color: kDarkGreyColor,
59              fontWeight: FontWeight.w600,
60            ), // TextStyle
61          ), // Text
62        ],
63      ); // Column
64    }
65  }

```

```

notification.dart • control_button.dart • default_button.dart •
lib > LandingScreen > components > default_button.dart > DefaultButton > build
1  import 'package:flutter/cupertino.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_auth/SensorScreen/sensor_screen.dart';
4  import 'package:flutter_auth/constants_2.dart';
5
6  class DefaultButton extends StatelessWidget {
7    const DefaultButton({
8      required Key key,
9      required this.size,
10     required this.title,
11     required this.press,
12   }) : super(key: key);
13
14     final Size size;
15     final String title;
16     final GestureTapCallback press;
17
18     @override
19     Widget build(BuildContext context) {
20       return GestureDetector(
21         onTap: () {
22           Navigator.push(
23             context,
24             CupertinoPageRoute(
25               builder: (context) => SensorScreen(),
26             ), // CupertinoPageRoute
27           );
28         },
29         child: Container(
30           height: size.height * 0.07,
31           width: double.infinity,
32           decoration: BoxDecoration(
33             color: kOrangeColor,
34             borderRadius: BorderRadius.circular(10),
35             boxShadow: [
36               BoxShadow(
37                 color: kOrangeColor.withOpacity(0.5),

```

```

38             blurRadius: 15,
39             offset: Offset(0, 12),
40           ), // BoxShadow
41           BoxShadow(
42             color: kOrangeColor.withOpacity(0.5),
43             blurRadius: 40,
44             offset: Offset(-3, -3),
45           ), // BoxShadow
46         ],
47       ), // BoxDecoration
48       child: Center(
49         child: Text(
50           'Next',
51           style: TextStyle(
52             color: Colors.black,
53             fontSize: 20,
54             fontWeight: FontWeight.w600,
55           ), // TextStyle
56         ), // Text
57       ), // Center
58     ), // Container
59   ); // GestureDetector
60 }
61 }

```

```

login_form.dart
lib > Screens > Login > components > login_form.dart > LoginForm
1  import 'package:flutter/material.dart';
2  import 'package:firebase_auth/firebase_auth.dart';
3  import 'package:firebase_core/firebase_core.dart';
4  import '../components/already_have_an_account_check.dart';
5  import '../constants.dart';
6  import '../Signup/signup_screen.dart';
7  import 'package:flutter_auth/SensorScreen/sensor_screen.dart';
8
9  final _formstate = GlobalKey<FormState>();
10 final user = FirebaseAuth.instance.currentUser;
11 //final uid = user;
12 String email = "";
13 String password = "";
14 //final auth = FirebaseAuth.instance;
15
16 void main() => runApp(LoginForm());
17
18 class LoginForm extends StatelessWidget {
19
20   @override
21   const LoginForm({
22     Key? key,
23   }) : super(key: key);
24
25   @override
26   Widget build(BuildContext context) {
27     return Form(
28       autovalidateMode: AutovalidateMode.always,
29       key: _formstate,
30       child: Column(
31         children: [
32           TextFormField(
33             textInputAction: TextInputAction.next,
34             cursorColor: kPrimaryColor,
35             onSaved: (value) {
36               email = value!.trim();

```

```

36               email = value!.trim();
37               //email.toString.trim();
38             },
39             decoration: InputDecoration(
40               hintText: "USERNAME",
41               prefixIcon: Padding(
42                 padding: const EdgeInsets.all(defaultPadding),
43                 child: Icon(Icons.person, size: 25),
44               ), // Padding
45             ), // InputDecoration
46           ), // TextFormField
47           Padding(
48             padding: const EdgeInsets.symmetric(vertical: defaultPadding),
49             child: TextFormField(
50               // controller: _passwordController,
51               textInputAction: TextInputAction.done,
52               obscureText: true,
53               cursorColor: kPrimaryColor,
54               onSaved: (value) {
55                 password = value!;
56                 //password.toString.trim();
57               },
58               decoration: InputDecoration(
59                 hintText: "PASSWORD",
60                 prefixIcon: Padding(
61                   padding: const EdgeInsets.all(defaultPadding),
62                   child: Icon(Icons.lock),
63                 ), // Padding
64               ), // InputDecoration
65             ), // TextFormField
66           ), // Padding
67           const SizedBox(height: defaultPadding),
68           Hero(
69             tag: "login_btn",

```

```

login_form.dart
lib > Screens > Login > components > login_form.dart > LoginForm

70   child: GestureDetector(
71     child: ElevatedButton(
72       onPressed: () async {
73         _formstate.currentState!.save();
74         print(email);
75         print(password);
76         try {
77           await Firebase.initializeApp();
78           final res = await FirebaseAuth.instance
79             .signInWithEmailAndPassword(
80               email: email, password: password);
81           print(res);
82           const snackBar = SnackBar(
83             content: Text('Login Success'),
84           ); // SnackBar
85           ScaffoldMessenger.of(context).showSnackBar(snackBar);
86           Navigator.pushAndRemoveUntil(
87             context,
88             MaterialPageRoute(
89               builder: (context) {
90                 // return TemperaturePage();
91                 return SensorScreen();
92               },
93             ), // MaterialPageRoute
94             (route) => false,
95           );
96         } catch (e) {
97           print('error');
98           print(e);
99           const snackBar = SnackBar(
100             content: Text('The username or password is incorrect !'),
101           ); // SnackBar
102           ScaffoldMessenger.of(context).showSnackBar(snackBar);
103         }
104       },
105       style: ElevatedButton.styleFrom(
106         backgroundColor: kPrimaryColor

```

```

106         backgroundColor: kPrimaryColor
107       ),
108       child: Text(
109         "Login".toUpperCase(),
110       ), // Text
111     ), // ElevatedButton
112   ), // GestureDetector
113 ), // Hero
114 const SizedBox(height: defaultPadding),
115 AlreadyHaveAnAccountCheck(
116   press: () {
117     Navigator.push(
118       context,
119       MaterialPageRoute(
120         builder: (context) {
121           return SignUpScreen();
122         },
123       ), // MaterialPageRoute
124     );
125   },
126 ), // AlreadyHaveAnAccountCheck
127 ],
128 ), // Column
129 ); // Form
130 }
131 }

```

```

login_form.dart • login_screen_top_image.dart X
lib > Screens > Login > components > login_screen_top_image.dart > LoginScreenTopImage > build
1 import 'package:flutter/material.dart';
2 import 'package:flutter_svg/flutter_svg.dart';
3
4 import '../../constants.dart';
5
6 class LoginScreenTopImage extends StatelessWidget {
7   const LoginScreenTopImage({
8     Key? key,
9   }) : super(key: key);
10  @override
11  Widget build(BuildContext context) {
12    return Column(
13      children: [
14        Text(
15          "Welcome!",
16          style: TextStyle
17            (fontFamily: "Montserrat",fontWeight: FontWeight.bold,)),
18        ), // Text
19        SizedBox(height: defaultPadding *0),
20        Row(
21          children: [
22            const Spacer(),
23            Expanded(
24              flex: 7,
25              child: SvgPicture.asset("assets/icons/Tabletlogin.svg"),
26            ), // Expanded
27            const Spacer(),
28          ],
29        ), // Row
30        SizedBox(height: defaultPadding * 0),
31      ],
32    ); // Column
33  }
34
35 }

```

```

login_form.dart • login_screen.dart •
lib > Screens > Login > login_screen.dart > LoginScreen > build
1 import 'package:flutter/material.dart';
2 import 'package:flutter_auth/responsive.dart';
3 import '../../components/background.dart';
4 import 'components/login_form.dart';
5 import 'components/login_screen_top_image.dart';
6
7 class LoginScreen extends StatelessWidget {
8   const LoginScreen({Key? key}) : super(key: key);
9
10  @override
11  Widget build(BuildContext context) {
12    return Background(
13      child: SingleChildScrollView(
14        child: Responsive(
15          mobile: const MobileLoginScreen(),
16          desktop: Row(
17            children: [
18              const Expanded(
19                child: LoginScreenTopImage(),
20              ), // Expanded
21              Expanded(
22                child: Row(
23                  mainAxisAlignment: MainAxisAlignment.center,
24                  children: const [
25                    SizedBox(
26                      width: 450,
27                      child: LoginForm(),
28                    ), // SizedBox
29                  ],
30                ), // Row
31            ], // Expanded
32          ), // Row
33        ), // Responsive
34      ), // SingleChildScrollView
35    ); // Background
36  }
37 }

```

```

38 }
39
40 class MobileLoginScreen extends StatelessWidget {
41   const MobileLoginScreen({
42     Key? key,
43   }) : super(key: key);
44
45   @override
46   Widget build(BuildContext context) {
47     return Column(
48       mainAxisAlignment: MainAxisAlignment.center,
49       children: <Widget>[
50         const LoginScreenTopImage(),
51         Row(
52           children: const [
53             Spacer(),
54             Expanded(
55               flex: 8,
56               child: LoginForm(),
57             ), // Expanded
58             Spacer(),
59           ],
60         ), // Row
61       ], // <Widget>[]
62     ); // Column
63   }
64 }

```

```

or_divider.dart
lib > Screens > Signup > components > or_divider.dart > OrDivider
1 import 'package:flutter/material.dart';
2 import 'package:flutter_auth/constants.dart';
3 class OrDivider extends StatelessWidget {
4   const OrDivider({Key? key}) : super(key: key);
5   @override
6   Widget build(BuildContext context) {
7     Size size = MediaQuery.of(context).size;
8     return Container(
9       margin: EdgeInsets.symmetric(vertical: size.height * 0.02),
10      width: size.width * 0.8,
11      child: Row(
12        children: <Widget>[
13          buildDivider(),
14          const Padding(
15            padding: EdgeInsets.symmetric(horizontal: 10),
16            child: Text(
17              "OR",
18              style: TextStyle(
19                color: kPrimaryColor,
20                fontWeight: FontWeight.w600,
21              ), // TextStyle
22            ), // Text
23          ), // Padding
24          buildDivider(),
25        ], // <Widget>[]
26      ), // Row
27    ); // Container
28  }
29  Expanded buildDivider() {
30    return const Expanded(
31      child: Divider(
32        color: Color(0xFFD9D9D9),
33        height: 1.5,
34      ), // Divider
35    ); // Expanded
36  }
37 }

```

```

or_divider.dart • sign_up_top_image.dart X
lib > Screens > Signup > components > sign_up_top_image.dart > ...
1 import 'package:flutter/material.dart';
2 import 'package:flutter_svg/flutter_svg.dart';
3 import '../constants.dart';
4
5 class SignUpScreenTopImage extends StatelessWidget {
6   const SignUpScreenTopImage({
7     Key? key,
8   }) : super(key: key);
9
10  @override
11  Widget build(BuildContext context) {
12    return Column(
13      children: [
14        Text(
15          "Sign Up".toUpperCase(),
16          style: TextStyle(fontFamily: "Montserrat",fontWeight: FontWeight.bold,)),
17        // Text
18        SizedBox(height: defaultPadding),
19        Row(
20          children: [
21            const Spacer(),
22            Expanded(
23              flex: 4,
24              child: SvgPicture.asset("assets/icons/Signup.svg"),
25            ), // Expanded
26            const Spacer(),
27          ],
28        ), // Row
29        SizedBox(height: defaultPadding),
30      ],
31    ); // Column
32  }
33 }

```

```

or_divider.dart • signup_form.dart •
lib > Screens > Signup > components > signup_form.dart > SignUpForm > build
1 import 'package:flutter/material.dart';
2 import 'package:firebase_auth/firebase_auth.dart';
3 import 'package:firebase_core/firebase_core.dart';
4 import '../components/already_have_an_account_check.dart';
5 import '../constants.dart';
6 import '../Login/login_screen.dart';
7
8 final _formstate = GlobalKey<FormState>();
9 String email = "";
10 String password = "";
11 // final auth = FirebaseAuth.instance;
12 class SignUpForm extends StatelessWidget {
13   const SignUpForm({
14     Key? key,
15   }) : super(key: key);
16
17   @override
18   Widget build(BuildContext context) {
19     return Form(
20       autovalidateMode: AutovalidateMode.always,
21       key: _formstate,
22       child: Column(
23         children: [
24           TextFormField(
25             keyboardType: TextInputType.emailAddress,
26             textInputAction: TextInputAction.next,
27             cursorColor: kPrimaryColor,
28             onSave: (value) {
29               email = value!.trim();
30             },
31             decoration: InputDecoration(
32               hintText: "USERNAME",
33               prefixIcon: Padding(
34                 padding: const EdgeInsets.all(defaultPadding),
35                 child: Icon(Icons.person),
36               ), // Padding
37             ), // InputDecoration

```

```

38     ), // TextFormField
39     Padding(
40       padding: const EdgeInsets.symmetric(vertical: defaultPadding),
41       child: TextFormField(
42         textInputAction: TextInputAction.done,
43         obscureText: true,
44         cursorColor: kPrimaryColor,
45         onSave: (value) {
46           password = value!.trim();
47         },
48         decoration: InputDecoration(
49           hintText: "PASSWORD",
50           prefixIcon: Padding(
51             padding: const EdgeInsets.all(defaultPadding),
52             child: Icon(Icons.lock),
53           ), // Padding
54         ), // InputDecoration
55       ), // TextFormField
56     ), // Padding
57     const SizedBox(height: defaultPadding / 2),
58     ElevatedButton(
59       onPressed: () async {
60         try{
61           _formstate.currentState!.save();
62           print(email);
63           print(password);
64           await Firebase.initializeApp();
65           final res = await FirebaseAuth.instance.createUserWithEmailAndPassword(email: email, password: password);
66           print(res);
67           const snackBar = SnackBar(
68             content: Text('Create account Success.'),
69           ); // SnackBar
70           ScaffoldMessenger.of(context).showSnackBar(snackBar);
71           Navigator.pop(context,);
72         }
73         catch (e){

```

```

74           print('Error');
75           print(e);
76           const snackBar = SnackBar(
77             content: Text('Can\'t create account.'),
78           ); // SnackBar
79           ScaffoldMessenger.of(context).showSnackBar(snackBar);
80         }
81       },
82       style: ElevatedButton.styleFrom(
83         backgroundColor: kPrimaryColor
84       ),
85       child: Text("Sign Up".toUpperCase()),
86     ), // ElevatedButton
87     const SizedBox(height: defaultPadding),
88     AlreadyHaveAnAccountCheck(
89       login: false,
90       press: () {
91         Navigator.push(
92           context,
93           MaterialPageRoute(
94             builder: (context) {
95               return LoginScreen();
96             },
97           ), // MaterialPageRoute
98         );
99       },
100     ), // AlreadyHaveAnAccountCheck
101   ],
102 ), // Column
103 ); // Form
104 }
105 }

```



```

social_icon.dart X
lib > Screens > Signup > components > social_icon.dart > Socallcon
1 import 'package:flutter/material.dart';
2 import 'package:flutter_auth/constants.dart';
3 import 'package:flutter_svg/flutter_svg.dart';
4
5 class SocialIcon extends StatelessWidget {
6   final String? iconSrc;
7   final Function? press;
8   const SocialIcon({
9     Key? key,
10    this.iconSrc,
11    this.press,
12  }) : super(key: key);
13
14  @override
15  Widget build(BuildContext context) {
16    return GestureDetector(
17      onTap: press as void Function()?,
18      child: Container(
19        margin: const EdgeInsets.symmetric(horizontal: 10),
20        padding: const EdgeInsets.all(20),
21        decoration: BoxDecoration(
22          border: Border.all(
23            width: 2,
24            color: kPrimaryLightColor,
25          ), // Border.all
26          shape: BoxShape.circle,
27        ), // BoxDecoration
28        child: SvgPicture.asset(
29          iconSrc!,
30          height: 20,
31          width: 20,
32        ), // SvgPicture.asset
33      ), // Container
34    ); // GestureDetector
35  }
36

```

```

signup_screen.dart
lib > Screens > Signup > signup_screen.dart > SignUpScreen > build
1 import 'package:flutter/material.dart';
2 import 'package:flutter_auth/constants.dart';
3 import 'package:flutter_auth/responsive.dart';
4 import '../components/background.dart';
5 import 'components/sign_up_top_image.dart';
6 import 'components/signup_form.dart';
7
8 class SignUpScreen extends StatelessWidget {
9   const SignUpScreen({Key? key}) : super(key: key);
10
11  @override
12  Widget build(BuildContext context) {
13    return Background(
14      child: SingleChildScrollView(
15        child: Responsive(
16          mobile: const MobileSignUpScreen(),
17          desktop: Row(
18            children: [
19              const Expanded(
20                child: SignUpScreenTopImage(),
21              ), // Expanded
22              Expanded(
23                child: Column(
24                  mainAxisAlignment: MainAxisAlignment.center,
25                  children: const [
26                    SizedBox(
27                      width: 450,
28                      child: SignUpForm(),
29                    ), // SizedBox
30                    SizedBox(height: defaultPadding / 2),
31                    // SocialSignUp()
32                  ],
33                ), // Column
34              ), // Expanded
35            ],
36          ), // Row
37        ), // Responsive
38      ),
39    );
40  }
41

```

```

38     ), // SingleChildScrollView
39   ); // Background
40 }
41 }
42 class MobileSignUpScreen extends StatelessWidget {
43   const MobileSignUpScreen({
44     Key? key,
45   }) : super(key: key);
46
47   @override
48   Widget build(BuildContext context) {
49     return Column(
50       mainAxisAlignment: MainAxisAlignment.center,
51       children: <Widget>[
52         const SignUpScreenTopImage(),
53         Row(
54           children: const [
55             Spacer(),
56             Expanded(
57               flex: 8,
58               child: SignUpForm(),
59             ), // Expanded
60             Spacer(),
61           ],
62         ), // Row
63         // const SocialSignUp()
64       ], // <Widget>[]
65     ); // Column
66   }
67 }

```

```

temperature.dart •
lib > Screens > temp > temperature.dart > _TemperaturePageState > Widget_roundedButton
1 import 'package:flutter/material.dart';
2 import 'package:percent_indicator/circular_percent_indicator.dart';
3
4 class TemperaturePage extends StatefulWidget {
5   const TemperaturePage({Key? key}) : super(key: key);
6
7   @override
8   _TemperaturePageState createState() => _TemperaturePageState();
9 }
10 class _TemperaturePageState extends State<TemperaturePage> {
11   double SPRINGER = 12;
12   double fan = 15;
13   bool light0 = true;
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       backgroundColor: Color.fromARGB(255, 249, 248, 246),
18       body: SafeArea(
19         child: Container(
20           margin: const EdgeInsets.only(top: 18, left: 24, right: 24),
21           child: Column(
22             mainAxisAlignment: MainAxisAlignment.start,
23             crossAxisAlignment: CrossAxisAlignment.start,
24             children: [
25               Row(
26                 mainAxisAlignment: MainAxisAlignment.spaceBetween,
27                 children: [
28                   GestureDetector(
29                     onTap: () {
30                       Navigator.pop(context);
31                     },
32                     child: const Icon(
33                       Icons.arrow_back_ios,
34                       color: Colors.indigo,
35                     ), // Icon
36                   ), // GestureDetector

```

```

36      ), // GestureDetector
37    ],
38  ), // Row
39  Expanded(
40    child: ListView(
41      physics: const BouncingScrollPhysics(),
42      children: [
43        const SizedBox(height: 32),
44        CircularPercentIndicator(
45          radius: 180,
46          lineWidth: 14,
47          percent: 0.75,
48          progressColor: Color.fromARGB(255, 142, 158, 128),
49          center: const Text(
50            '26\u00B0',
51            style: TextStyle(
52              fontSize: 32,
53              fontWeight: FontWeight.bold,
54            ), // TextStyle
55          ), // Text
56        ), // CircularPercentIndicator
57        const SizedBox(height: 24),
58        const Center(
59          child: Text(
60            'TEMPERATURE',
61            style: TextStyle(
62              fontWeight: FontWeight.bold, color: Colors.black54), // TextStyle
63          ), // Text
64        ), // Center
65        const SizedBox(height: 24),
66        Container(
67          alignment: Alignment.center,
68          child: Column(
69            // crossAxisAlignment: CrossAxisAlignment.stretch,
70            children: [

```

```

71          ],
72        ), // Container
73      ], // Row
74    ), // Container
75    Container(
76      width: 400,
77      padding: const EdgeInsets.symmetric(vertical: 24),
78      decoration: BoxDecoration(
79        color: Colors.white,
80        borderRadius: BorderRadius.circular(8),
81      ), // BoxDecoration
82      child: Column(
83        crossAxisAlignment: CrossAxisAlignment.center,
84        children: <Widget>[
85          const Padding(
86            padding: EdgeInsets.symmetric(horizontal: 10),
87            child: Text(
88              'SPRINGER',
89              style: TextStyle(
90                fontWeight: FontWeight.bold,
91              ), // TextStyle
92            ), // Text
93          ), // Padding
94          Switch(
95            value: light0,
96            activeColor: Color.fromARGB(255, 142, 158, 128),
97            onChanged: (bool value){
98              setState(() {
99                light0 = value;
100              });
101            }
102          ), // Switch
103        ], // <Widget>[]
104      ), // Column
105    ), // Container
106    Container(

```

```

107 width: 400,
108 padding: const EdgeInsets.symmetric(vertical: 24),
109 decoration: BoxDecoration(
110   color: Colors.white,
111   borderRadius: BorderRadius.circular(8),
112 ), // BoxDecoration
113 child: Column(
114   crossAxisAlignment: CrossAxisAlignment.center,
115   children: <Widget>[
116     const Padding(
117       padding: EdgeInsets.symmetric(horizontal: 10),
118       child: Text(
119         'SPRINGER',
120         style: TextStyle(
121           fontWeight: FontWeight.bold,
122         ), // TextStyle
123       ), // Text
124     ), // Padding
125     Switch(
126       value: light0,
127       activeColor: Color.fromARGB(255, 142, 158, 128),
128       onChanged: (bool value){
129         setState(() {
130           light0 = value;
131         });
132       }
133     ), // Switch
134   ], // <Widget>[]
135 ), // Column
136 ), // Container
137 const SizedBox(width: 24),
138 ]
139 ), // Row
140 ), // Container
141 const SizedBox(height: 24),
142 Container(
143   child: Row(

```

```

144 mainAxisAlignment: MainAxisAlignment.spaceBetween,
145 children: [
146   Container(
147     width: 400,
148     padding: const EdgeInsets.symmetric(vertical: 24),
149     decoration: BoxDecoration(
150       color: Colors.white,
151       borderRadius: BorderRadius.circular(8),
152     ), // BoxDecoration
153     child: Column(
154       crossAxisAlignment: CrossAxisAlignment.center,
155       children: <Widget>[
156         const Padding(
157           padding: EdgeInsets.symmetric(horizontal: 10),
158           child: Text(
159             'SPRINGER',
160             style: TextStyle(
161               fontWeight: FontWeight.bold,
162             ), // TextStyle
163           ), // Text
164         ), // Padding
165         Switch(
166           value: light0,
167           activeColor: Color.fromARGB(255, 142, 158, 128),
168           onChanged: (bool value){
169             setState(() {
170               light0 = value;
171             });
172           }
173         ), // Switch
174       ], // <Widget>[]
175     ), // Column
176   ), // Container
177   Container(
178     width: 400,
179     padding: const EdgeInsets.symmetric(vertical: 24),

```

```

login_signup_btn.dart
lib > Screens > Welcome > components > login_signup_btn.dart > LoginAndSignupBtn
1  import 'package:flutter/material.dart';
2  import '../constants.dart';
3  import '../Login/login_screen.dart';
4  import '../Signup/signup_screen.dart';
5
6  class LoginAndSignupBtn extends StatelessWidget {
7    const LoginAndSignupBtn({
8      Key? key,
9    }) : super(key: key);
10
11  @override
12  Widget build(BuildContext context) {
13    return Column(
14      children: [
15        Hero(
16          tag: "login_btn",
17          child: ElevatedButton(
18            onPressed: () {
19              Navigator.push(
20                context,
21                MaterialPageRoute(
22                  builder: (context) {
23                    return LoginScreen();
24                  },
25                ), // MaterialPageRoute
26              );
27            },
28            style: ElevatedButton.styleFrom(
29              backgroundColor: kPrimaryColor
30            ),
31            child: Text(
32              "Login".toUpperCase(),
33              style: TextStyle(fontFamily: "Montserrat"),
34            ), // Text
35          ), // ElevatedButton
36        ), // Hero
37        const SizedBox(height: 16),

```

```

37        const SizedBox(height: 16),
38        ElevatedButton(
39          onPressed: () {
40            Navigator.push(
41              context,
42              MaterialPageRoute(
43                builder: (context) {
44                  return SignUpScreen();
45                },
46              ), // MaterialPageRoute
47            );
48          },
49          style: ElevatedButton.styleFrom(
50            backgroundColor: kPrimaryLightColor, elevation: 0,
51          ),
52          child: Text(
53            "Sign Up".toUpperCase(),
54            style: TextStyle(color: Colors.black, fontFamily: "Montserrat"),
55          ), // Text
56        ), // ElevatedButton
57      ],
58    ); // Column
59  }
60

```

```

login_signup_btn.dart • welcome_image.dart •
lib > Screens > Welcome > components > welcome_image.dart > WelcomelImage > build
1  import 'package:flutter/material.dart';
2  import 'package:flutter_svg/flutter_svg.dart';
3  import '../constants.dart';
4
5  class WelcomeImage extends StatelessWidget {
6    const WelcomeImage({
7      Key? key,
8    }) : super(key: key);
9    @override
10   Widget build(BuildContext context) {
11     return Column(
12       children: [
13         Text(
14           "\n\n\nwelcome to",
15           style: TextStyle(
16             fontFamily: "Montserrat",
17             fontWeight: FontWeight.bold
18           ), // TextStyle
19         ), // Text
20         Text(
21           "'MUSHROOM FARM'",
22           style: TextStyle(
23             fontFamily: "Montserrat",
24             fontWeight: FontWeight.bold,
25             fontSize: 25
26           ), // TextStyle
27         ), // Text
28         SizedBox(height: defaultPadding * 0.0),
29         Row(
30           children: [
31             Spacer(),
32             Expanded(
33               flex: 7,
34               child: SvgPicture.asset(
35                 "assets/icons/welcome1.svg",
36               ), // SvgPicture.asset
37             ), // Expanded
38           ],
39         ), // Row
40         SizedBox(height: defaultPadding * 1),
41       ],
42     ); // Column
43   }
44 }
45

```

```

welcome_screen.dart
lib > Screens > Welcome > welcome_screen.dart > WelcomeScreen > build
1  import 'package:flutter/material.dart';
2  import '../components/background.dart';
3  import '../components/responsive.dart';
4  import 'components/login_signup_btn.dart';
5  import 'components/welcome_image.dart';
6
7  class WelcomeScreen extends StatelessWidget {
8    const WelcomeScreen({Key? key}) : super(key: key);
9    @override
10   Widget build(BuildContext context) {
11     return Background(
12       child: SingleChildScrollView(
13         child: SafeArea(
14           child: Responsive(
15             desktop: Row(
16               mainAxisAlignment: MainAxisAlignment.end,
17               children: [
18                 const Expanded(
19                   child: WelcomeImage(),
20                 ), // Expanded
21                 Expanded(
22                   child: Row(
23                     mainAxisAlignment: MainAxisAlignment.center,
24                     children: const [
25                       SizedBox(
26                         width: 450,
27                         child: LoginAndSignupBtn(),
28                       ), // SizedBox
29                     ],
30                   ), // Row
31                 ), // Expanded
32               ],
33             ), // Row
34             mobile: const MobileWelcomeScreen(),
35           ), // Responsive
36         ), // SafeArea
37       ), // SingleChildScrollView

```

```

38     ); // Background
39   }
40 }
41 class MobileWelcomeScreen extends StatelessWidget {
42   const MobileWelcomeScreen({
43     Key? key,
44   }) : super(key: key);
45
46   @override
47   Widget build(BuildContext context) {
48     return Column(
49       mainAxisAlignment: MainAxisAlignment.center,
50       children: <Widget>[
51         const WelcomeImage(),
52         Row(
53           children: const [
54             Spacer(),
55             Expanded(
56               flex: 8,
57               child: LoginAndSignupBtn(),
58             ), // Expanded
59             Spacer(),
60           ],
61         ), // Row
62       ], // <Widget>[]
63     ); // Column
64   }
65 }

```

```

body.dart •
lib > SensorScreen > components > body.dart > getSensorData
1  import 'dart:core';
2  import 'package:firebase_storage/firebase_storage.dart';
3  import 'package:flutter/material.dart';
4  import 'package:flutter_auth/Screens/Welcome/welcome_screen.dart';
5  import 'package:flutter_auth/SensorScreen/components/cam.dart';
6  import 'package:flutter_auth/SensorScreen/components/image_card.dart';
7  import 'package:flutter_auth/SensorScreen/components/sensor_display.dart';
8  import 'package:flutter_auth/constants_2.dart';
9  import 'package:firebase_database/firebase_database.dart';
10 import 'package:intl/intl.dart';
11
12 late String TEMP = '';
13 late String HUMI = '';
14 late double GAS = 0;
15 var value;
16
17 getSensorData ()async{
18   FirebaseDatabase database = FirebaseDatabase.instance;
19   await database.ref('Data/ValueSensor/Gas').onValue.listen((DatabaseEvent event){
20     final data = event.snapshot.value;
21     // print(data);
22     try{
23       GAS = double.parse(double.parse(data.toString()).toStringAsFixed(2));
24       print(GAS);
25     }
26     catch(e){
27
28     }
29     // gas = int.parse(data.toString());
30     // print(gas);
31   });
32   await database.ref('Data/ValueSensor/Humidity').onValue.listen((DatabaseEvent event){
33     // print(data);
34     try{
35       final data = event.snapshot.value;
36       HUMI = data.toString();
37       print(HUMI);

```

```

38   }
39   catch (e){
40     HUMI = "ERR";
41   }
42   // gas = int.parse(data.toString());
43   // print(gas);
44 });
45 await database.ref('Data/ValueSensor/Temperature').onValue.listen((DatabaseEvent event){
46   final data = event.snapshot.value;
47   // print(data);
48   try{
49     TEMP = data.toString();
50     print(TEMP);
51   }
52   catch (e){
53
54   }
55   // gas = int.parse(data.toString());
56   // print(gas);
57 });
58 await FirebaseStorage.instance.ref('pictures/image.jpg').getDownloadURL();
59 }
60
61 class SensorScreenBody extends StatefulWidget {
62   @override
63   _SensorScreenBodyState createState() => _SensorScreenBodyState();
64 }
65
66 class _SensorScreenBodyState extends State<SensorScreenBody> {
67
68
69   DateTime myDateTime = DateTime.now();
70
71   @override
72   Widget build(BuildContext context) {
73     Size size = MediaQuery.of(context).size;

```



```

74   var formatter = DateFormat.yMMMMd('en_US');
75   return SafeArea(
76     child: Padding(
77       padding: EdgeInsets.symmetric(horizontal: size.width * 0.05),
78       child: ListView(
79         children: [
80           SizedBox(height: size.height * 0.02),
81           Row(
82             mainAxisAlignment: MainAxisAlignment.spaceBetween,
83             children: [
84               SizedBox(),
85               Text(
86                 'Home',
87                 style: TextStyle(
88                   fontFamily: "Montserrat",fontWeight: FontWeight.bold,
89                   color: Colors.fromARGB(221, 0, 0, 0),
90
91                   fontSize: 20,
92                 ), // TextStyle
93             ], // Text
94           GestureDetector(
95             onTap: () => Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => WelcomeScreen())),
96             child: Container(
97               height: size.height * 0.045,
98               width: size.width * 0.095,
99               decoration: BoxDecoration(
100                 color: kBgColor,
101                 borderRadius: BorderRadius.circular(30),
102                 boxShadow: [
103                   BoxShadow(
104                     color: Colors.fromARGB(31, 51, 75, 11),
105                     blurRadius: 8,
106                     offset: Offset(3, 3),
107                   ), // BoxShadow
108                 ],
109               ), // BoxDecoration

```

```

109             ), // BoxDecoration
110             child: Icon(
111               Icons.exit_to_app_outlined,
112               color: kGreenColor,
113             ), // Icon
114           ], // Container
115         ), // GestureDetector
116       ],
117     ), // Row
118     SizedBox(height: size.height * 0.03),
119     Row(
120       children: [
121         Container(
122           ), // Container
123         SizedBox(width: size.width * 0.040),
124         Column(
125           crossAxisAlignment: CrossAxisAlignment.start,
126           children: [
127             Text(
128               '${formatter.format(myDateTime)}',
129               style: TextStyle(
130                 color: Colors.grey,
131                 fontWeight: FontWeight.w600,
132                 fontFamily: "Montserrat",
133               ), // TextStyle
134             ), // Text
135             // title: Text(data.Title),
136             // subtitle: Text(DateFormat("dd/MM/yyyy").format(date.date)),
137             Text(
138               'Hi!\nMushroom Farm',
139               style: TextStyle(
140                 color: Colors.fromARGB(221, 0, 0, 0),
141                 fontWeight: FontWeight.bold,
142                 fontSize: 30,
143               ), // TextStyle
144             ), // Text

```

```

146         ],
147       ), // Column
148     ],
149   ), // Row
150   SizedBox(height: size.height * 0.030),
151   Row(
152     mainAxisAlignment: MainAxisAlignment.spaceBetween,
153     children: [
154       SensorDisplay(title: "TEMP", url: "Data/ValueSensor/Temperature", unit: "°C"),
155       SensorDisplay(title: "HUM", url: "Data/ValueSensor/Humidity", unit: "%"),
156       SensorDisplay(title: "GAS", url: "Data/ValueSensor/Gas", gas: true)
157     ],
158   ), // Row
159
160   SizedBox(height: size.height * 0.050),
161   Row(
162     mainAxisAlignment: MainAxisAlignment.spaceAround,
163     children: [
164       CustomCard(
165         key: UniqueKey(),
166         size: size,
167         icon: Icon(
168           Icons.wind_power_outlined,
169           size: 35,
170           color: Colors.grey.shade400,
171         ), // Icon
172         title: "HEATER",
173         statusOn: "On",
174         statusOff: "Off",
175         url: "Data/Status/HeaterStatus",
176       ), // CustomCard
177       // Switch(
178       //   value: switchState,
179       //   onChanged: onSwitchChanged,
180     ),
181     CustomCard(
182       key: UniqueKey(),

```

```

182       key: UniqueKey(),
183       size: size,
184       icon: Icon(
185         Icons.water_drop_rounded,
186         size: 35,
187         color: Colors.grey.shade400,
188       ), // Icon
189       title: "SPRINGER",
190       statusOn: "Spray",
191       statusOff: "Off",
192       url: "Data/Status/WaterPumpStatus",
193     ), // CustomCard
194   ],
195   ), // Row
196   SizedBox(height: size.height * 0.050),
197   ImageCard(),
198 ],
199 ), // ListView
200 ), // Padding
201 ); // SafeArea
202 }
203 }
204 }
205

```

```

body.dart • cam.dart •
lib > SensorScreen > components > cam.dart > CustomCard
1  import 'package:firebase_database/firebase_database.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_auth/constants_2.dart';
4
5  class CustomCard extends StatefulWidget {
6    const CustomCard(
7      {
8        required Key key,
9        required this.size,
10       required this.icon,
11       required this.title,
12       required this.statusOn,
13       required this.statusOff,
14       required this.url,
15     })
16     : super(key: key);
17
18     final Size size;
19     final Icon icon;
20     final String title;
21     final String statusOn;
22     final String statusOff;
23     final String url;
24
25     @override
26     _CustomCardState createState() => _CustomCardState();
27   }
28   class _CustomCardState extends State<CustomCard>
29     with SingleTickerProviderStateMixin {
30     late AnimationController _animationController;
31     late Animation<Alignment> _animation;
32     bool isChecked = true;
33     String type = "";
34
35     FirebaseDatabase database = FirebaseDatabase.instance;

```

```

38   void initState() {
39     database.ref(widget.url).onValue.listen((event) {
40       type = event.snapshot.value.toString().split(" ")[0];
41       String data = event.snapshot.value.toString().split(" ")[1];
42       setState(() {
43         isChecked = data == "On";
44       });
45     });
46     _animationController = AnimationController(
47       vsync: this,
48       duration: Duration(milliseconds: 200),
49     ); // AnimationController
50
51     _animation = Tween<Alignment>(
52       begin: Alignment.bottomCenter, end: Alignment.topCenter
53     ).animate(
54       CurvedAnimation(
55         parent: _animationController,
56         curve: Curves.easeOutCubic,
57       ),
58     );
59     super.initState();
60   }
61
62   @override
63   Widget build(BuildContext context) {
64     if (isChecked) {
65       _animationController.animateTo(0);
66     } else {
67       _animationController.animateTo(20);
68     }
69     return Container(
70       height: 140,
71       width: widget.size.width * 0.35,
72       decoration: BoxDecoration(
73         color: kBgColor,

```

```

body.dart • cam.dart
lib > SensorScreen > components > cam.dart > CustomCard

74   borderRadius: BorderRadius.circular(20),
75   boxShadow: [
76     BoxShadow(
77       color: Colors.black12,
78       blurRadius: 8,
79       offset: Offset(3, 3),
80     ), // BoxShadow
81     BoxShadow(
82       color: Color.fromARGB(255, 235, 229, 223),
83       blurRadius: 0,
84       offset: Offset(-3, -3),
85     ), // BoxShadow
86   ],
87 ), // BoxDecoration
88 child: Padding(
89   padding: const EdgeInsets.all(15),
90   child: Column(
91     crossAxisAlignment: CrossAxisAlignment.start,
92     children: [
93       Row(
94         mainAxisAlignment: MainAxisAlignment.spaceBetween,
95         children: [
96           widget.icon,
97           widget.title != "LEAKS"
98             ? AnimatedBuilder(
99               animation: _animationController,
100              builder: (animation, child) {
101                return GestureDetector(
102                  onTap: () {
103                    if (type.isEmpty) { return; }
104                    database.ref("Data/Status/").update(
105                      type == "Pump" ? {
106                        "WaterPumpStatus": "Pump On"
107                      } : {
108                        "HeaterStatus": "Heater On"
109                      }
110                    ).then((value) => null);

```

```

111   ],
112   child: Container(
113     height: 40,
114     width: 25,
115     decoration: BoxDecoration(
116       borderRadius: BorderRadius.circular(20),
117       color: Color.fromARGB(255, 235, 235, 235),
118       boxShadow: [
119         BoxShadow(
120           color: Colors.grey.shade200,
121           blurRadius: 8,
122           offset: Offset(3, 3),
123         ), // BoxShadow
124         BoxShadow(
125           color: Colors.black12,
126           blurRadius: 5,
127           offset: Offset(-3, -3),
128         ), // BoxShadow
129       ],
130     ), // BoxDecoration
131     child: Align(
132       alignment: _animation.value,
133       child: Container(
134         width: 15,
135         height: 15,
136         margin: EdgeInsets.symmetric(
137           vertical: 2, horizontal: 1), // EdgeInsets.symmetric
138         decoration: BoxDecoration(
139           color: isChecked
140             ? Color.fromARGB(255, 103, 82, 82)
141             : kGreenColor,
142           shape: BoxShape.circle,
143         ), // BoxDecoration
144       ), // Container
145     ), // Align
146   ), // container

```

```

146         ), // Container
147       ); // GestureDetector
148     },
149   ) // AnimatedBuilder
150   : Container(),
151 ],
152 ), // Row
153 SizedBox(height: 10),
154 Text(
155   widget.title,
156   style: TextStyle(
157     fontSize: 16,
158     fontWeight: FontWeight.bold,
159     color: kBlueColor,
160   ), // TextStyle
161 ), // Text
162 Text(
163   !isChecked ? widget.statusOff : widget.statusOn,
164   style: TextStyle(
165     fontWeight: FontWeight.bold,
166     color: !isChecked ? Colors.grey.withOpacity(0.6) : kGreenColor,
167   ), // TextStyle
168 ), // Text
169 ],
170 ), // Column
171 ), // Padding
172 ); // Container
173 }
174 }

```

```

body.dart • cam.dart • image_card.dart •
lib > SensorScreen > components > image_card.dart > _ImageCardState > getImageUrl
1  import 'package:flutter/material.dart';
2  import 'package:flutter_auth/constants_2.dart';
3  import 'package:firebase_storage/firebase_storage.dart';
4
5  class ImageCard extends StatefulWidget {
6    const ImageCard({Key? key}) : super(key: key);
7
8    @override
9    State<ImageCard> createState() => _ImageCardState();
10 }
11
12 class _ImageCardState extends State<ImageCard> {
13   FirebaseStorage storage = FirebaseStorage.instance;
14   String url = "";
15   String date = "";
16
17   double height = 250;
18
19   getImageUrl() async {
20     url = "";
21     setState(() {});
22     var ref = storage.ref("/pictures").child("image.jpg");
23     url = await ref.getDownloadURL();
24     var data = await ref.getMetadata();
25     date = data.updated.toString();
26     setState(() {
27       // :)
28     });
29   }
30
31   @override
32   void initState() {
33     getImageUrl();
34     super.initState();
35   }
36
37   @override
38   Widget build(BuildContext context) {
39     Size size = MediaQuery.of(context).size;

```

```

38   return Container(
39     clipBehavior: Clip.antiAlias,
40     height: height,
41     width: size.width * 0.8,
42     decoration: BoxDecoration(
43       color: kBgColor,
44       borderRadius: BorderRadius.circular(20),
45       boxShadow: [
46         BoxShadow(
47           color: Colors.black12,
48           blurRadius: 8,
49           offset: Offset(3, 3),
50         ), // BoxShadow
51         BoxShadow(
52           color: Color.fromARGB(255, 235, 235, 229, 223),
53           blurRadius: 0,
54           offset: Offset(-3, -3),
55         ), // BoxShadow
56       ],
57     ), // BoxDecoration
58     child: Stack(
59       children: [
60         if (url.isNotEmpty) ... [
61           Image.network(url, width: size.width, height: height, fit: BoxFit.fitWidth),
62           Positioned(
63             top: 16,
64             right: 16,
65             child: GestureDetector(
66               onTap: () {
67                 getImageUrl();
68               },
69               child: Container(
70                 height: size.height * 0.045,
71                 width: size.width * 0.095,
72                 decoration: BoxDecoration(
73                   color: kBgColor,

```

```

74         borderRadius: BorderRadius.circular(30),
75         boxShadow: [
76           BoxShadow(
77             color: Color.fromARGB(31, 51, 75, 11),
78             blurRadius: 8,
79             offset: Offset(3, 3),
80           ), // BoxShadow
81         ],
82       ), // BoxDecoration
83       child: Icon(
84         Icons.refresh,
85         color: kGreenColor,
86       ), // Icon
87     ), // Container
88   ), // GestureDetector
89 ), // Positioned
90 ), // Positioned
91 Positioned(
92   right: 16,
93   bottom: 8,
94   child: Text(date, style: TextStyle(color: Color.fromARGB(200, 140, 140, 140))),
95 ), // Positioned
96 ] else ... [
97   Center(
98     child: CircularProgressIndicator(
99       color: kGreenColor,
100     ), // CircularProgressIndicator
101   ), // Center
102 ],
103 ],
104 ), // Stack
105 ); // Container
106 }
107 }

```

```

sensor_display.dart X
lib > SensorScreen > components > sensor_display.dart > _SensorDisplayState > initState
1  import 'package:firebase_database/firebase_database.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_auth/components/notification.dart';
4  import 'package:flutter_auth/constants_2.dart';
5
6  class SensorDisplay extends StatefulWidget {
7    final String title;
8    final String url;
9    final String unit;
10   final bool gas;
11   const SensorDisplay({
12     required this.title,
13     required this.url,
14     this.unit = "",
15     this.gas = false,
16     Key? key
17   }) : super(key: key);
18
19   @override
20   State<SensorDisplay> createState() => _SensorDisplayState();
21 }
22
23 class _SensorDisplayState extends State<SensorDisplay> {
24   FirebaseDatabase db = FirebaseDatabase.instance;
25   String data = "";
26
27   @override
28   void initState() {
29     db.ref(widget.url).onValue.listen((event) {
30       if (event.snapshot.value.toString() != "NaN") {
31         data = (event.snapshot.value as double).toStringAsFixed(1);
32         setState(() {
33           // ;)
34         });
35         if (widget.gas) {
36           double d = double.parse(data);
37           if (d > 1000) {

```

```

sensor_display.dart X
lib > SensorScreen > components > sensor_display.dart > _SensorDisplayState > initState
38     NotificationService.showNotification(
39       title: "การแจ้งเตือน",
40       body: "ปริมาณแก๊สสูงเกินไป"
41       // TODO: here ;)
42     );
43   }
44 }
45
46 });
47 super.initState();
48 }
49 @override
50 Widget build(BuildContext context) {
51   Size size = MediaQuery.of(context).size;
52   return Container(
53     padding: EdgeInsets.all(8.0),
54     height: size.height * 0.12,
55     width: size.height * 0.12,
56     decoration: BoxDecoration(
57       color: kBgColor,
58       borderRadius: BorderRadius.circular(100),
59       boxShadow: [
60         BoxShadow(
61           color: Colors.black12,
62           blurRadius: 8,
63           offset: Offset(3, 3),
64         ), // BoxShadow
65         BoxShadow(
66           color: Color.fromARGB(255, 235, 235, 229),
67           blurRadius: 0,
68           offset: Offset(-3, -3),
69         ), // BoxShadow
70       ],
71     ), // BoxDecoration
72     child: Column(
73       mainAxisAlignment: MainAxisAlignment.max,

```

```

74     children: [
75       Text(
76         widget.title,
77         style: TextStyle(
78           fontSize: 14,
79           fontWeight: FontWeight.bold,
80           color: Colors.black,
81         ), // TextStyle
82       ), // Text
83       SizedBox(height: size.height * 0.12 * 0.15,),
84       Text(
85         data.isEmpty ? "loading..." :
86         "$data ${widget.unit}",
87         style: TextStyle(
88           fontSize: 14,
89           fontWeight: FontWeight.bold,
90           color: Colors.black,
91         ), // TextStyle
92       ), // Text
93     ],
94   ), // Column
95 ); // Container
96 }
97 }

```

sensor_screen.dart ✕

```

lib > SensorScreen > sensor_screen.dart > ...
1  import 'package:flutter/material.dart';
2  import 'package:flutter_auth/SensorScreen/components/body.dart';
3  import 'package:flutter_auth/constants_2.dart';
4
5  class SensorScreen extends StatelessWidget {
6    @override
7    Widget build(BuildContext context) {
8      return Scaffold(
9        backgroundColor: kBgColor,
10       body: SensorScreenBody(),
11     ); // Scaffold
12   }
13 }

```

constants_2.dart ✕

```

lib > constants_2.dart > ...
1  import 'dart:ui';
2
3  const Color kBgColor = Color.fromARGB(255, 235, 229, 223);
4  const Color kOrangeColor = Color(0xFFF07662);
5  const Color kDarkGreyColor = Color(0xFF727C9B);
6  const Color kGreenColor = Color.fromARGB(255, 142, 158, 128);
7  const Color kBlueColor = Color(0xFF07062e);
8

```



```

180         decoration: BoxDecoration(
181           color: Colors.white,
182           borderRadius: BorderRadius.circular(8),
183         ), // BoxDecoration
184         child: Column(
185           crossAxisAlignment: CrossAxisAlignment.center,
186           children: <Widget>[
187             const Padding(
188               padding: EdgeInsets.symmetric(horizontal: 10),
189               child: Text(
190                 'SPRINGER',
191                 style: TextStyle(
192                   fontWeight: FontWeight.bold,
193                 ), // TextStyle
194               ), // Text
195             ), // Padding
196             Switch(
197               value: light0,
198               activeColor: Color.fromARGB(255, 142, 158, 128),
199               onChanged: (bool value){
200                 setState(() {
201                   light0 = value;
202                 });
203               }
204             ), // Switch
205           ], // <Widget>[]
206         ), // Column
207       ), // Container
208       const SizedBox(width: 24),
209     ], // Row
210   ), // Container
211 ], // Container
212 ], // Column
213 ], // Container
214 ], // ListView
215 ], // ListView
216 ], // ListView

```

```

217 ], // Expanded
218 ], // Column
219 ], // Container
220 ], // SafeArea
221 ], // Scaffold
222 ); // Scaffold
223 }
224 widget_fan({
225   required String title,
226   bool isActive = false,
227 }) {
228   return Column(
229     children: [
230       Container(
231         padding: const EdgeInsets.all(18),
232         decoration: BoxDecoration(
233           color: isActive ? Color.fromARGB(255, 160, 141, 119) : Colors.white,
234           borderRadius: BorderRadius.circular(18),
235         ), // BoxDecoration
236         child: Image.asset(
237           isActive ? '../assets/images/fan-2.png' : '../assets/images/fan-1.png',
238         ), // Image.asset
239       ), // Container
240       const SizedBox(height: 12),
241       Text(
242         title,
243         style: TextStyle(
244           color: isActive ? Colors.black87 : Colors.black54,
245         ), // TextStyle
246       ), // Text
247     ],
248   ); // Column
249 }
250 widget_roundedButton({
251   required String title,
252   bool isActive = false,

```

```

253   }) {
254     return Container(
255       padding: const EdgeInsets.symmetric(
256         vertical: 12,
257         horizontal: 32,
258       ), // EdgeInsets.symmetric
259       decoration: BoxDecoration(
260         color: isActive ? Color.fromARGB(255, 142, 141, 119) : Colors.transparent,
261         borderRadius: BorderRadius.circular(24),
262         border: Border.all(color: Color.fromARGB(255, 152, 159, 155)),
263       ), // BoxDecoration
264       child: Text(
265         title,
266         style: TextStyle(
267           color: isActive ? Colors.white : Colors.black,
268         ), // TextStyle
269       ), // Text
270     ); // Container
271   }
272 }
273

```

```

lib > constants.dart > ...
1  import 'package:flutter/material.dart';
2
3  const kPrimaryColor = Color.fromARGB(255, 142, 158, 128);
4  const kPrimaryLightColor = Color.fromARGB(255, 235, 229, 223);
5  const double defaultPadding = 16.0;

```



```

37         elevation: 0,
38         shape: const StadiumBorder(),
39         maximumSize: const Size(double.infinity, 56),
40         minimumSize: const Size(double.infinity, 56),
41       ),
42     ),
43     ), // ElevatedButtonThemeData
44     inputDecorationTheme: const InputDecorationTheme(
45       filled: true,
46       fillColor: kPrimaryLightColor,
47       iconColor: kPrimaryColor,
48       prefixIconColor: kPrimaryColor,
49       contentPadding: EdgeInsets.symmetric(
50         horizontal: defaultPadding, vertical: defaultPadding), // EdgeInsets.symmetric
51       border: OutlineInputBorder(
52         borderRadius: BorderRadius.all(Radius.circular(30)),
53         borderSide: BorderSide.none,
54       ), // OutlineInputBorder
55     )), // InputDecorationTheme // ThemeData
56     home: WelcomeScreen(),
57   ); // MaterialApp
58 }
59 }
60
61 class NoScrollGlow extends ScrollBehavior {
62   @override
63   Widget buildOverscrollIndicator(
64     BuildContext context, Widget child, ScrollableDetails details) {
65     return child;
66   }
67 }

```

```

constants.dart • main.dart • responsive.dart X
lib > responsive.dart > Responsive > isMobile
1 import 'package:flutter/material.dart';
2
3 class Responsive extends StatelessWidget {
4   final Widget mobile;
5   final Widget? tablet;
6   final Widget desktop;
7
8   const Responsive({
9     Key? key,
10    required this.mobile,
11    this.tablet,
12    required this.desktop,
13  }) : super(key: key);
14
15   static bool isMobile(BuildContext context) =>
16     MediaQuery.of(context).size.width < 576;
17
18   static bool isTablet(BuildContext context) =>
19     MediaQuery.of(context).size.width >= 576 &&
20     MediaQuery.of(context).size.width <= 992;
21
22   static bool isDesktop(BuildContext context) =>
23     MediaQuery.of(context).size.width > 992;
24
25   @override
26   Widget build(BuildContext context) {
27     final Size size = MediaQuery.of(context).size;
28     if (size.width > 992) {
29       return desktop;
30     } else if (size.width >= 576 && tablet != null) {
31       return tablet!;
32     } else {
33       return mobile;
34     }
35   }
36 }

```

ภาคผนวก ค

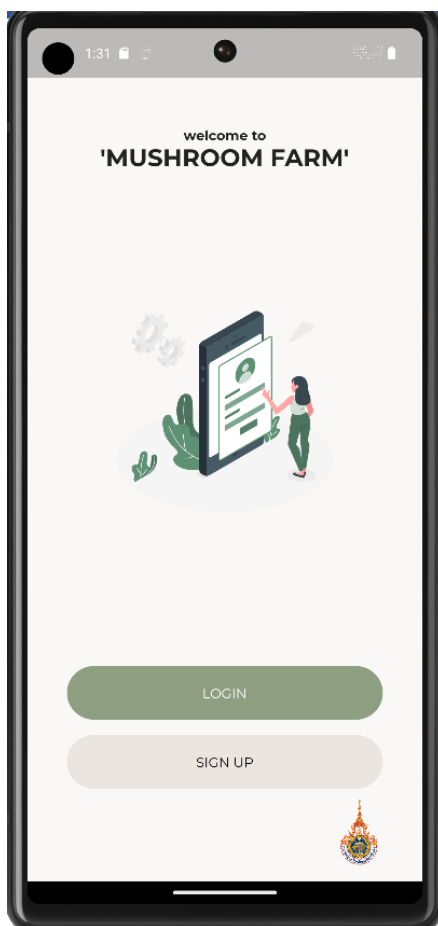
คู่มือการใช้งานแอปพลิเคชัน

เนื้อหาภาคผนวก ค

1. หน้าหลักแอปพลิเคชัน จะมีปุ่ม Login และ Sign up

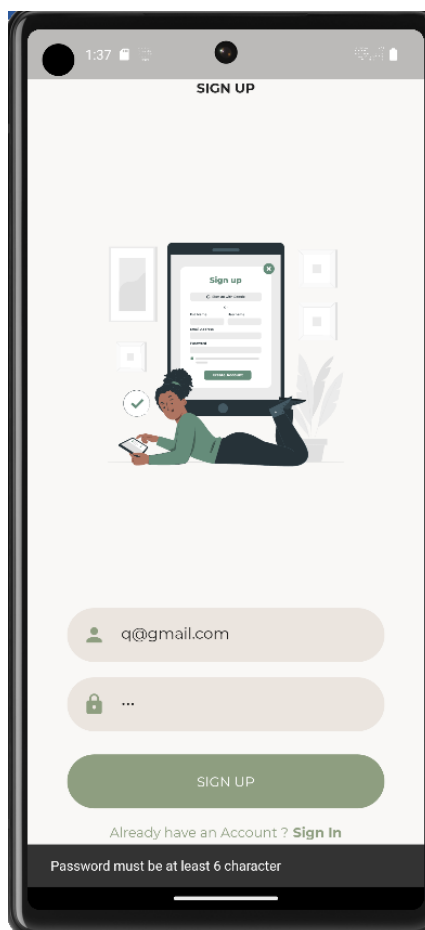
1.1 Sign Up – กดปุ่มลงทะเบียนสำหรับผู้ที่ยังไม่มีบัญชี

1.2 Log In - กดปุ่มเข้าสู่ระบบเพื่อเข้าสู่ระบบสำหรับผู้ที่เคยลงทะเบียนแล้ว



2.หน้า Sign Up (ลงทะเบียน)

2.1 การลงทะเบียนแนะนำให้ใช้อีเมลในการลงทะเบียน และใส่รหัสผ่านมากกว่า 6 ตัว



1:37

SIGN UP

Sign up

q@gmail.com

...

SIGN UP

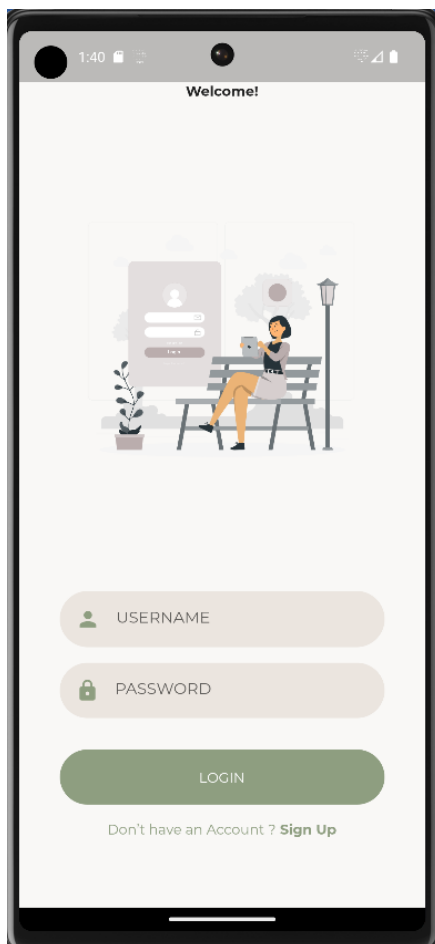
Already have an Account ? [Sign In](#)

Password must be at least 6 character

2.2 หรือถ้ายังไม่มีบัญชี สามารถกดคำว่า Sign In ด้านล่างได้

3. หน้า Log In (หน้าลงชื่อเข้าใช้ระบบ)

3.1 ลงชื่อเข้าใช้ด้วยอีเมลและรหัสผ่านที่ได้ลงทะเบียนไว้



4. หน้าสำหรับดูค่าต่างๆ

4.1 แสดงวันที่ด้านบน

4.2 ฟังก์ชัน TEMP ไว้สำหรับแสดงอุณหภูมิ

4.3 ฟังก์ชัน HUM ไว้สำหรับแสดงความชื้น

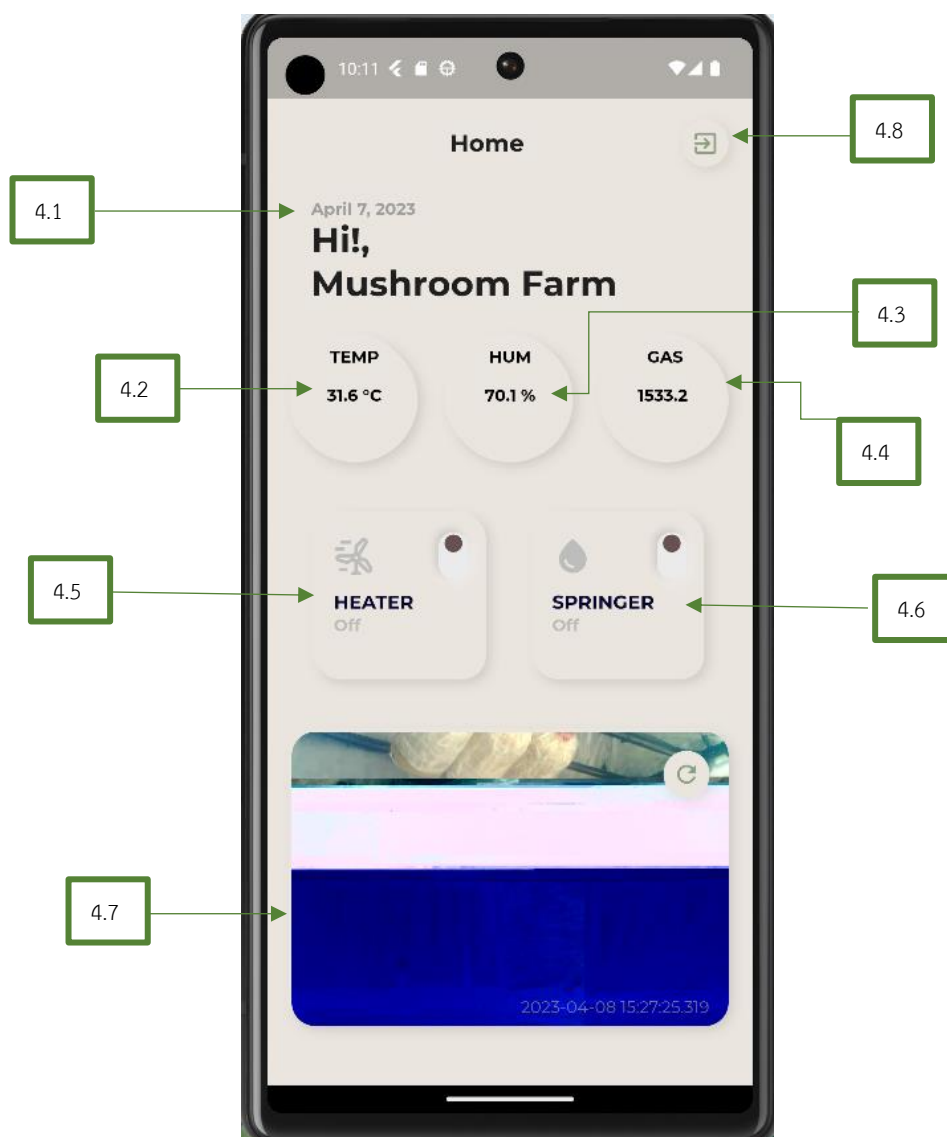
4.4 ฟังก์ชัน GAS ไว้สำหรับแสดงค่าก๊าซ

4.5 ฟังก์ชัน HEATER ไว้สำหรับแสดงสถานะเปิดปิดของฮีตเตอร์

4.6 ฟังก์ชัน SPRINGER ไว้สำหรับแสดงสถานะเปิดปิดของสปริงเกอร์

4.7 กล้องภายในโรงเรือน

4.8 ปุ่มออกเพื่อกลับไปหน้าจอหลัก



ภาคผนวก ง

คู่มือการติดตั้งระบบภายในโรงเรียนเพาะเห็ดนางฟ้าภูฐาน

เนื้อหาภาคผนวก ง

คู่มือการติดตั้งระบบภายในโรงเรือนเพาะเห็ดนางฟ้าภูฐาน

ระบบที่ใช้ภายในโรงเรือนจะมีการติดตั้งอุปกรณ์ต่างๆ โดยจะแบ่งออก ดังนี้

1. ส่วนของอุปกรณ์เซนเซอร์และอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ที่ใช้ในโรงเรือนเพาะเห็ดนางฟ้าภูฐาน อุปกรณ์เซนเซอร์และอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ภายในโรงเรือนจะประกอบไปด้วย
 - 1.1 เซนเซอร์วัดคุณภาพอากาศ รุ่น MQ-135 ผู้จัดทำจะทำการต่อกับบอร์ดไมโครคอนโทรเลอร์ รุ่น ESP8266
 - 1.2 เซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ รุ่น DHT11 ผู้จัดทำจะทำการต่อกับบอร์ดไมโครคอนโทรเลอร์ รุ่น ESP8266
 - 1.3 กล้องที่ติดตั้งภายในโรงเรือนเพาะเห็ดนางฟ้าภูฐาน จะเป็นรุ่น OV2640 ผู้จัดทำจะทำการต่อกับบอร์ดไมโครคอนโทรเลอร์ รุ่น ESP32 และติดตั้งมาพร้อมกับโมดูลอัพโหลด
 - 1.4 รีเลย์แบบ 2 ช่อง ขนาด 5v ใช้ในการควบคุมอุปกรณ์ให้ทำงานได้แบบอัตโนมัติ
 - 1.5 เพาเวอร์ซัพพลาย ขนาด 12v 5A ใช้สำหรับเป็นแหล่งจ่ายไฟ
 - 1.6 โมดูลขับเคลื่อนมอเตอร์ รุ่น L298N ใช้สำหรับควบคุมการหมุนของพัดลม
 - 1.7 วงจรลดแรงดันไฟฟ้า (Step Down) ใช้สำหรับลดแรงดันไฟฟ้าที่ส่งมาจากแหล่งจ่ายไฟ
2. ส่วนของอุปกรณ์พ่นหมอกและระบบปั้มน้ำที่ใช้ในโรงเรือนเพาะเห็ดนางฟ้าภูฐาน
 - 2.1 ปั้มน้ำเดี่ยวขนาด 10.8 บาร์ ทางเข้าน้ำและทางออกน้ำจะต้องใช้สายยางขนาด 8-8.5 มิลลิเมตร มอเตอร์ปั้มสามารถทำงานต่อเนื่องได้ 3-4 ชั่วโมง ตัวปั้มน้ำจะใช้ไฟ DC 12v โดยผู้จัดทำจะทำการต่อเข้ากับเพาเวอร์ซัพพลาย 12v ในการจ่ายไฟให้กับปั้มน้ำและปั้มน้ำจะมีการต่อกับกับรีเลย์ขนาด 5v เพื่อเป็นการทำให้ปั้มน้ำสามารถทำงานได้แบบอัตโนมัติ
 - 2.2 หัวพ่นหมอกทองเหลือง ขนาด 1.2 มิลลิเมตร สามารถหมุนปรับละอองของน้ำได้ ควรใช้กับปั้มน้ำที่มีแรงดัน 2-3 บาร์ เพื่อความเป็นละอองที่ดี
3. ส่วนของพัดลมระบายอากาศ

3.1 พัฒนาระบายความร้อนขนาด 12 เซนติเมตร แรงดันไฟ DC 12v 0.5A ทิศทางการ
หมุนของพัดลมจะหมุนทวนเข็มนาฬิกาจากทิศทางของใบพัดลม

วงจรการต่อระบบภายในโรงเรือนเพาะเห็ดนางฟ้าภูฐาน

บอร์ดไมโครคอนโทรลเลอร์ ESP8266

A0 ต่อเข้ากับ output MQ-135
D2 ต่อเข้ากับ output DHT11
D5 ต่อเข้ากับ in2 relay
D6 ต่อเข้ากับ in1 relay
D7 ต่อเข้ากับ ENB Motor drive
GND ต่อเข้ากับ Fotoboard V-
3V ต่อเข้ากับ Fotoboard V+

โมดูลขับมอเตอร์ (Motor drive)

In3 ต่อเข้ากับ Fotoboard V+
In4 ต่อเข้ากับ Fotoboard V-
ENB ต่อเข้ากับ D7
Out3 ต่อเข้ากับ fan01
Out4 ต่อเข้ากับ fan02
GND ต่อเข้ากับ Fotoboard V- และ power supply V-

แหล่งจ่ายไฟ (Power supply)

+V ต่อเข้ากับ IN+ Step down และ relay COM2
-V ต่อเข้ากับ GND motor drive , Pump V- และ IN- Step down
L ต่อเข้ากับ Heater V+
N ต่อเข้ากับ relay COM1

Relay

NO2 ต่อเข้ากับ Pump V+
COM2 ต่อเข้ากับ power supply V+

ประวัติผู้จัดทำปริญญานิพนธ์

ประวัติผู้จัดทำปริญญานิพนธ์



ชื่อ	นางสาวรุจิมาส ปิยมงคลชัย
สาขาวิชา/ภาควิชา	สาขาเทคโนโลยีคอมพิวเตอร์
วัน-เดือน-ปี เกิด	วันที่ 30 สิงหาคม 2544
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
ที่อยู่	81/266 ซ.8 หมู่บ้านฟ้าปิยรมณ (นิวา) หมู่ 4 อำเภอลำลูกกา ตำบลบึงคำพร้อย จังหวัดปทุมธานี
ประวัติการศึกษา	มัธยมศึกษาตอนปลาย (ม.6) โรงเรียนสตรีวัดมหาพฤฒาราม ในพระบรมราชินูปถัมภ์ ปีที่สำเร็จการศึกษา 2562

ประวัติผู้จัดทำปริญญานิพนธ์



ชื่อ	นางสาวศุภิสรา เกิดแจ่ม
สาขาวิชา/ภาควิชา	สาขาเทคโนโลยีคอมพิวเตอร์
วัน-เดือน-ปี เกิด	วันที่ 27 กุมภาพันธ์ พ.ศ.2540
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
ที่อยู่	147/24 หมู่บ้านมณีนรินทร์พาร์ค ตำบลบ้านกลาง อำเภอเมือง จังหวัด ปทุมธานี
ประวัติการศึกษา	มัธยมศึกษาตอนปลาย (ม.6) โรงเรียนโพธิธรรมสุวัฒน์ ปีที่สำเร็จการศึกษา 2560