

ระบบควบคุมโรงเรือนสามารถฟาร์มแบบปิด

SMART FARM CONTROL SYSTEM

ชนพร ศุภะศร

ครุต คงจะคงคง

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาชีวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

ปริญญาอินชีปการศึกษา 2562

ภาควิชาช่างกรรมคอมพิวเตอร์

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ระบบควบคุมฟาร์มอัจฉริยะแบบบีด

SMART FARM CONTROL SYSTEM

ผู้จัดทำ

1. นายชนพร ศุภะศร รหัสนักศึกษา 60015023

2. นายศรุต คงชนะพงศ์ รหัสนักศึกษา 60015044

อาจารย์ที่ปรึกษา

(รศ.ดร.เจริญ วงศ์ชุมเนิน)

ระบบควบคุมฟาร์มอัจฉริยะแบบปิด

นายธนพร	ศุภระศร	60015023
นายศรุต	คณะคณะพงศ์	60015044
รศ.ดร.เจริญ	วงศ์ชุมເຊັນ	อาจารย์ที่ปรึกษา
ปีการศึกษา 2562		

บทคัดย่อ

ปัจจุบัน Internet of Things (IoT) เป็นเรื่องที่ได้รับความนิยมอย่างมาก โดย IoT หมายถึง สิ่งของต่าง ๆ ที่มีการเชื่อมต่อกันเป็นเครือข่าย สามารถสื่อสารรับส่งข้อมูลถึงกันได้อัตโนมัติ หรือ ทำงานได้เอง โดยไม่ต้องอาศัยมนุษย์ งานวิจัยนี้ จึงได้เป็นการนำเสนอระบบจัดการดูแลฟาร์ม อัจฉริยะแบบปิดที่สามารถเก็บข้อมูลการทำงาน เป็นการใช้งานราสเตอร์พาย (Raspberry Pi) เป็นตัวควบคุมการทำงานของระบบ โดยควบคุมการทำงานของแสงในการเพาะปลูก (LED Grow Light) ควบคุมการทำความเย็นด้วยเซนเซอร์วัดค่าอุณหภูมิและความชื้นในอากาศ (DHT22) และ ควบคุมการดน้ำด้วยเซนเซอร์วัดค่าความชื้นในดิน (Soil Moisture Sensor) แล้วยังสามารถควบคุม การทำงานของระบบนี้ ได้จากระยะไกลด้วยเว็บแอปพลิเคชัน สร้างความสะดวกสบายและง่ายต่อ การดูแล

กิตติกรรมประกาศ

ปริญญาบัณฑิตนี้สำเร็จคุณล่วงไประเป็นอย่างดี ด้วยคำแนะนำ คำปรึกษาและคอบขุ่นจาก
หลายๆ ฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษาที่ให้โอกาสเข้ามาเพื่อทำปริญญาบัณฑิตนี้โดย
เอ้าใจใส่ ให้คำแนะนำและความช่วยเหลือเสมอมา คือ รศ.ดร.จริญ วงศ์ชุมยืน ซึ่งเป็นอาจารย์ที่
ปรึกษา ต้องขอบพระคุณเป็นอย่างสูง

ขอขอบคุณอาจารย์ทุกท่านในภาควิชาศึกษาการคอมพิวเตอร์ที่ให้ความรู้ ประสบการณ์
ต่างๆ ในการทำกิจกรรมและเพื่อนๆ ที่ให้กำลังใจตลอดเวลาที่เรียนในสถาบันเทคโนโลยีพระจอม
เกล้าเจ้าคุณทหารลาดกระบังตลอดจนสั่งสอนสิ่งที่ดีเสมอมา

สุดท้ายนี้ขอขอบคุณภาควิชาศึกษาการคอมพิวเตอร์ที่ได้จัดเตรียมสิ่งอำนวยความสะดวก
เพื่อให้การวิจัยและพัฒนาเว็บแอปพลิเคชันเป็นไปได้ด้วยความสะดวกและรวดเร็ว รวมทั้งยังมี
อินเทอร์เน็ตความเร็วสูงให้บริการ สำหรับการค้นคว้าหาข้อมูล ความรู้ต่างๆ ที่สำคัญและจำเป็นใน
การทำนิยงานให้เสร็จคุณล่วงการดำเนินการดำเนินงานในรายงานฉบับนี้สำเร็จคุณล่วงไปได้ด้วยดี

ชนพร ศุภารักษ์

ครุต คงจะคงพงศ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
สารบัญ.....	III
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
 บทที่ 1 บทนำ.....	 1
1.1 ความเป็นมาของปัญหา.....	2
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 ขอบเขตโครงการ.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
 บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	 3
2.1 ความรู้เบื้องต้นการเขียนโปรแกรมบนอินเทอร์เน็ต.....	3
2.1.1 หลักการทำงานของ WWW.....	3
2.1.2 ประเภทของโปรแกรมบนเว็บ.....	6
2.2 การติดตั้งโปรแกรม XAMPP.....	8
2.2.1 ขั้นตอนการติดตั้งโปรแกรม XAMPP.....	8
2.2.2 ระบบการจัดเก็บไฟล์ของโปรแกรม XAMPP.....	13
2.3 HTML Basic Element & Format.....	15
2.3.1 ลักษณะของแท็กและอิมเมนต์.....	15
2.3.2 แอ็ตทริบิวต์ของแท็ก.....	16
2.3.3 โครงสร้างหลักของ HTML.....	17
2.3.4 แท็กสำหรับหัวข้อและเอกสาร.....	18

สารบัญ (ต่อ)

	หน้า
2.3.5 แท็กสำหรับกำหนดขนาดหัวข้อ.....	20
2.3.6 แท็กสำหรับการแสดงผลส่วนของข้อความ.....	20
2.3.7 แท็กสำหรับจัดรูปแบบข้อความ.....	21
2.3.8 อักษรและพิเศษของ HTML.....	22
2.3.9 แท็กสำหรับแสดงคำอธิบาย.....	23
2.3.10 การเชื่อมโยงแบบต่างๆ.....	23
2.3.11 แท็กสำหรับแสดงรายการ.....	23
2.4 CSS Basic & Selector.....	25
2.4.1 พื้นฐานของ CSS.....	25
2.4.2 ขอบเขตของการกำหนดสไตล์แบบต่างๆ.....	26
2.4.3 รูปแบบซึ่งเลือกโดยที่น่าสนใจ.....	30
2.5 JavaScript Basic.....	34
2.5.1 คำสั่งของภาษาสคริปต์.....	35
2.5.2 องค์ประกอบเบื้องต้นของการเขียนโค้ด.....	35
2.6 PHP Basic and MySQL.....	40
2.6.1 การสร้างฟังก์ชันของ PHP.....	41
2.6.2 คำสั่งแสดงผลลัพธ์.....	42
2.6.3 คำสั่งหมายเหตุ (Comment).....	44
2.6.4 เรื่องของตัวแปรและค่าคงที่.....	45
2.6.5 การสร้างฟังก์ชันของ PHP.....	47
2.6.6 ฟังก์ชันของ PHP.....	48
2.6.7 Session และ Cookie.....	49
2.6.8 ฐานข้อมูล MySQL.....	49
2.6.9 การจัดการฐานข้อมูล MySQL ด้วย PhpMyAdmin.....	51
2.6.9 การเขียน PHP จัดการ MySQL.....	65

สารบัญ (ต่อ)

	หน้า
2.7 Python.....	67
2.8 Raspberry Pi.....	70
2.8.1 DHT22 Temperature and Humidity Sensor.....	72
2.8.2 Ambient Light Sensor Module GY-302 (BH1750FVI).....	74
2.8.3 Capacitive Analog Soil Moisture Sensor.....	76
2.8.4 จอ LCD 16x4 Character.....	77
2.8.5 LED Grow Light.....	79
2.8.6 Switching Power Supply.....	80
 บทที่ 3 การออกแบบและการพัฒนา.....	 82
3.1 ภาพรวมการทำงานของระบบ.....	82
3.2 การออกแบบตู้จำลองโรงเรือนแบบปิด.....	83
3.2.1 วงจรอ่านค่าจากเซ็นเซอร์ต่าง ๆ	84
3.2.2 วงจรควบคุมเอาต์พุต	85
3.2.3 การต่อจอย LCD 16*4.....	86
3.3 ส่วนฐานข้อมูล และ เว็บแอปพลิเคชัน.....	87
3.3.1 Entity User.....	88
3.3.2 Entity settings.....	88
3.3.3 Entity activity.....	88
3.3.4 Entity logs.....	89
3.3.5 Entity timers.....	89
3.3.6 การออกแบบเว็บแอปพลิเคชัน.....	90
 บทที่ 4 การทดลองและการทดสอบ.....	 94
4.1 สร้างตู้จำลองโรงเรือนแบบปิด.....	94

สารบัญ (ต่อ)

	หน้า
4.1.1 วัตถุประสงค์.....	94
4.1.2 วิธีการดำเนิน.....	94
4.1.3 ผลการทำงาน.....	95
4.2 ทดลองการเข้าใช้งาน Wed application.....	95
4.2.1 วัตถุประสงค์.....	95
4.2.2 วิธีการดำเนิน.....	95
4.2.3 ผลการทำงาน.....	96
4.3 ทดลองการส่งข้อมูลและสั่งงานผ่านอินเตอร์เน็ตด้วย Web application.....	96
4.3.1 วัตถุประสงค์.....	96
4.3.2 วิธีการดำเนิน.....	97
4.3.3 ผลการทำงาน.....	97
4.4 ทดลองการอ่านข้อมูลต่าง ๆ ข้อมูลหลังจาก web application.....	98
4.3.1 วัตถุประสงค์.....	98
4.3.2 วิธีการดำเนิน.....	98
4.3.3 ผลการทำงาน.....	98
4.5 ทดลองการรับน้ำและปรับอุณหภูมิแบบอัตโนมัติ.....	98
4.3.1 วัตถุประสงค์.....	98
4.3.2 วิธีการดำเนิน.....	98
4.3.3 ผลการทำงาน.....	99
4.6 ทดลองเพาะปลูกผักสดในครัวเรือนจำลอง.....	100
4.3.1 วัตถุประสงค์.....	100
4.3.2 วิธีการดำเนิน.....	100
4.3.3 ผลการทำงาน.....	100

สารบัญ (ต่อ)

	หน้า
บทที่ 5 บทสรุปและข้อเสนอแนะ.....	101
5.1 ผลการดำเนินงาน.....	101
5.2 ปัญหา อุปสรรค และแนวทางการแก้ไข.....	101
5.3 แนวทางการพัฒนา.....	102
บรรณานุกรม.....	103

สารบัญตาราง

ตาราง	หน้า
2.1 รายละเอียดของการประการฟังก์ชัน PHP.....	47
2.2 ฟังก์ชันเกี่ยวกับสตริง PHP.....	48
2.3 ฟังก์ชันเกี่ยวกับคณิตศาสตร์ PHP.....	49
2.4 ฟังก์ชันเกี่ยวกับวันเวลา PHP.....	49
2.5 คำสั่งเพื่อดำเนินการกับฐานข้อมูล.....	51
2.6 แสดงรายชื่อโมดูลมาตราฐานของ Python.....	69
2.7 คุณสมบัติของ Raspberry pi 3 Model B.....	69
2.8 รายละเอียดของ LED Grow Light ขนาด 10 W 12V.....	80
2.9 รายละเอียดของ Switching Power Supply ขนาด 12V 40A.....	81

สารบัญ

หัวข้อ	หน้า
2.1 แสดงการทำงานของ Client-Server.....	3
2.2 แสดงการทำงานของ ไคลเอนต์-เซิร์ฟเวอร์ ในลักษณะของการให้บริการงานพิมพ์.....	4
2.3 แสดงการทำงานของ ไคลเอนต์-เซิร์ฟเวอร์ ในลักษณะของการให้บริการฐานข้อมูล.....	4
2.4 แสดงการทำงานของ WWW.....	5
2.5 แสดงการทำงานของ Static Programming.....	6
2.6 แสดงการทำงานของ Dynamic Programming.....	7
2.7 แสดงหน้าเว็บ XAMPP Apache.....	8
2.8 เริ่มต้นการติดตั้ง โปรแกรม XAMPP.....	9
2.9 หน้าจอการเลือก Components ที่ต้องการติดตั้ง.....	9
2.10 หน้าจอการกำหนด ไดรฟ์ และ โฟลเดอร์ สำหรับการติดตั้ง.....	10
2.11 หน้าจอแสดงการตระเตรียม โปรแกรม Drupal, Joomla, World-press.....	11
2.12 หน้าจอ ก่อนเริ่มทำการติดตั้ง.....	11
2.13 หน้าจอแสดงระหว่างทำการติดตั้ง.....	12
2.14 หน้าจอแสดงการติดตั้งที่เสร็จสมบูรณ์.....	13
2.15 หน้าจอ Control Panel.....	13
2.16 แสดง โฟลเดอร์ และ ไฟล์ ที่จัดเก็บ.....	14
2.17 ลักษณะของแท็ก.....	15
2.18 แนวทางการกำหนดหัวข้อด้วย <hx>.....	20
2.19 แนวทางการใช้แท็กสำหรับการแบ่งส่วนข้อความ.....	21
2.20 การใช้งานแท็กสำหรับจัดรูปแบบข้อความ.....	22
2.21 อักษรพิเศษของ HTML.....	22
2.22 แนวทางการกำหนดสไลต์แบบ Inline.....	27
2.23 แนวทางการกำหนดสไลต์แบบ Internal.....	28
2.24 การเชื่อมโยงด้วยแท็ก <link>.....	29

สารบัญ (ต่อ)

หัวข้อ	หน้า
2.25 การเชื่อมโยงคีย์เวิร์ด @import.....	29
2.26 ชีลีกเตอร์แบบ Tag.....	31
2.27 ชีลีกเตอร์แบบ ID.....	32
2.28 ชีลีกเตอร์แบบ Class.....	33
2.29 คำส่วนของภาษาสคริปต์.....	35
2.30 การแสดงข้อความแจ้งเตือนด้วย alert().....	37
2.31 การแสดงข้อมูลบนหน้าเพจด้วย document.write().....	38
2.32 ภาษา PHP แทรกใน html.....	41
2.33 ภาษา html แทรกใน PHP.....	41
2.34 บรรทัดแสดงผลลัพธ์ของคำสั่ง echo.....	42
2.35 บรรทัดแสดงความผิดพลาดจากคำสั่ง print.....	43
2.36 การสร้าง และเรียกใช้ function ของภาษา php.....	48
2.37 การเรียกใช้ฐานข้อมูล MySQL ใน XAMPP.....	50
2.38 การเปิดใช้งานโปรแกรม PhpMyAdmin.....	52
2.39 กำหนดรหัสอักขระ utf8mb4_unicode_ci หรือ utf8_unicode_ci.....	52
2.40 Create database.....	53
2.41 หน้าจอหลังจากสร้างฐานข้อมูล.....	54
2.42 การตั้งชื่อตารางและกำหนดจำนวนฟิลด์.....	54
2.43 หน้าจอการกำหนดรายละเอียดของตาราง.....	55
2.44 กำหนดรายละเอียดของตาราง.....	56
2.45 หน้าจอสรุประยุทธ์ของตาราง.....	57
2.46 ตัวอย่างการแก้ไขรายละเอียดของฟิลด์ user_id.....	57
2.47 ผลลัพธ์หลังจากการแก้ไขรายละเอียดของฟิลด์ user_id.....	58
2.48 สำหรับการเพิ่มเรคอร์ดใหม่ในตาราง.....	59
2.49 หน้าจอการเพิ่มเรคอร์ดใหม่.....	59

สารบัญ (ต่อ)

หัวข้อ	หน้า
2.50 ผลลัพธ์การเพิ่ม rekordใหม่.....	60
2.51 ลิงก์ Browse สำหรับการคูดข้อมูลในตาราง.....	61
2.52 หน้าจอการแก้ไข rekord.....	61
2.53 หน้าจอแสดง Dialogue Box ยืนยันการลบ.....	62
2.54 ลิงก์ Drop สำหรับลบตาราง.....	63
2.55 หน้าจอยืนยันการลบตาราง.....	63
2.56 ลิงก์ Export สำหรับการส่งออกฐานข้อมูล.....	64
2.57 หน้าจอแสดงการเลือกไฟล์ฐานข้อมูลที่จะนำเข้า.....	65
2.58 แสดงสถาปัตยกรรมความสามารถของภาษา Python.....	67
2.59 โมดูลหลักที่เป็นมาตรฐานและ โมดูลที่ได้เพิ่มเติมให้กับ Python.....	69
2.60 ตำแหน่งของอุปกรณ์ต่าง ๆ บนบอร์ด Pi3.....	70
2.61 ตำแหน่งของ GPIO 40 PIN.....	71
2.62 DHT22 Temperature and Humidity Sensor.....	72
2.63 การต่อเซ็นเซอร์DHT22 กับ Raspberry pi.....	73
2.64 ค่าอุณหภูมิและความชื้นในอากาศที่วัดได้.....	73
2.65 Ambient Light Sensor Module GY-302.....	74
2.66 การเชื่อมต่อ BH1750FV กับ Raspberry pi.....	75
2.67 ค่าความเข้มแสงที่วัดได้.....	76
2.68 Capacitive Analog Soil Moisture Sensor.....	76
2.69 จอ LCD แบบ16*4.....	77
2.70 การเชื่อมต่อ BH1750FV กับ Raspberry pi.....	78
2.71 Address I ² C ของ LCD.....	78
2.72 แสดงข้อความบนจอ LCD.....	79
2.73 ลักษณะของ LED Grow Light ขนาด 10 W 12V.....	80
2.74 ลักษณะของ Switching Power Supply.....	81

สารบัญ (ต่อ)

หัวข้อ	หน้า
3.1 ภาพรวมของระบบควบคุม โครงเรือนสมาร์ทฟาร์มแบบปิด.....	82
3.2 ภาพแสดงการส่วนการทำงานของระบบ โครงเรือนสมาร์ทฟาร์มแบบปิด.....	83
3.3 การต่อวงจรอ่านค่าเซ็นเซอร์.....	84
3.4 การต่อวงจรควบคุมอาต์พุต.....	85
3.5 การต่อจอ LCD เข้ากับ Raspberry pi.....	86
3.6 data base Schema.....	87
3.7 Use case Diagram web application.....	90
3.8 ภาพแสดงหน้าจอเข้าสู่ระบบ.....	91
3.9 ภาพแสดงหน้าจอกล้อง.....	91
3.10 ภาพแสดงหน้าจอตั้งค่าการทำงาน.....	92
3.11 ภาพแสดงหน้าจอแสดงผลการทำงาน.....	92
3.12 ภาพแสดงหน้าจອการขัดการผู้ใช้งาน.....	93
4.1 ตู้จำลอง โครงเรือนแบบปิด.....	95
4.2 การเพิ่มUserในส่วนของผู้ดูแล.....	96
4.3 กรณีไม่มีUser หรือ กรอกข้อมูลผิด.....	96
4.4 หน้าweb application ที่ใช้ควบคุม ผ่านสมาร์ทโฟน.....	97
4.5 สถานะหน้าตู้เปลี่ยนตามที่ควบคุมผ่าน wed application.....	97
4.6 ข้อมูลต่าง ๆ ย้อนหลังบน web application.....	98
4.7 หน้าเมนูต่าง ๆ บน web application.....	99
4.8 หน้ารายการที่กำหนดการทำงานต่างๆ ไว้.....	99
4.9 การปลูกผักในตู้จำลอง โครงเรือน.....	100

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในอดีตประเทศไทยเป็นเมืองเกษตรที่มีความมั่งคั่งในเรื่องทรัพยากรทางด้านอาหาร มีความอุดมสมบูรณ์ เนื่องจากสภาพภูมิประเทศ มีความเหมาะสมในการทำเกษตรกรรมพืชพันธุ์ชั้นเยี่ยม สามารถหาได้จากธรรมชาติรอบตัว แต่ปัจจุบันสภาพแวดล้อมที่เคยเป็นธรรมชาติที่สมบูรณ์เริ่มมีการเสื่อมถอยจากการพัฒนาประเทศที่เน้นอุตสาหกรรมหนักอันก่อให้เกิดสภาวะโลกร้อนทำให้สภาพภูมิอากาศเปลี่ยนแปลงไป ส่งผลกระทบต่อสิ่งแวดล้อมและกิจกรรมด้านการเกษตร

ทางผู้วิจัยได้เล็งเห็นว่าการเปลี่ยนจากเกษตรกลางแจ้ง (Outdoor Farming) ซึ่งเป็นเกษตรแบบดั้งเดิมที่ต้องต่อสู้กับสภาพดินฟ้าอากาศสู่เกษตรในร่ม (Indoor Farming) ที่ทำการเพาะปลูกในสิ่งปลูกสร้างที่มีการควบคุมสภาพแวดล้อม เช่น การทำไวร์ในอาคารสูง (Vertical Farming) เพื่อเป็นแหล่งผลิตอาหารได้เอง

เมื่อระบบการเกษตรเปลี่ยนผ่านสู่ยุคสมัยของเกษตรอัจฉริยะหรือสมาร์ทฟาร์ม (Smart Farm) เพื่อให้มีความสามารถในการรับรู้ความเปลี่ยนแปลงต่างๆ ด้วยระบบเซ็นเซอร์ (sensor) หรืออุปกรณ์ตรวจจับสัญญาณหรือบิรุณทางพิสิกส์ต่างๆ เช่น อุณหภูมิ เสียง แสง การสัมผัส เป็นต้น ในการควบคุมการผลิตอาหารในปริมาณมากต่อประชากรโลกที่จะมีจำนวนมากขึ้นในอนาคต เกษตรกรและบุคลากรทางการเกษตรจะให้ความสำคัญกับการทำฟาร์มที่มีความแม่นยำสูง (Precision Farming) มีการใช้ทรัพยากรที่มีอยู่ให้คุ้มค่าที่สุดด้วยการดูแลทุกกระบวนการอย่างมีประสิทธิภาพและแม่นยำ ผ่านระบบเซ็นเซอร์ที่จะทำการเก็บข้อมูลและวิเคราะห์ด้วยเครื่องมือสมัยใหม่เพื่อให้กระบวนการผลิตถูกต้อง ตั้งแต่เริ่มหว่านเมล็ด จนถึงเก็บปุย ให้ทราบพื้นที่ที่เหมาะสมที่สุด การเก็บเกี่ยวและคัดเลือกผลผลิต เพื่อให้ได้ผลผลิตสูงสุด

งานวิจัยนี้จึงได้พัฒนาระบบควบคุมฟาร์มอัจฉริยะแบบปิดที่สามารถเก็บข้อมูลการทำงาน เป็นการใช้งานราสเตเบอร์พาย (Raspberry Pi) เป็นตัวควบคุมการทำงานของระบบ โดยควบคุมการทำงานของแสงในสิ่งปลูก (LED Grow Light) ควบคุมการทำความเย็นด้วยเซนเซอร์วัดค่าอุณหภูมิ และความชื้นในอากาศ (DHT22) และควบคุมการ湿润ด้วยเซนเซอร์วัดค่าความชื้นในดิน (Soil Moisture Sensor) เป็นต้น และยังสามารถควบคุมการทำงานของระบบน้ำได้จากระบบไกลด์วีบ แอพพลิเคชัน สร้างความสะดวกสบายและง่ายต่อการดูแล

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อศึกษาการทำงานระบบควบคุมฟาร์มอัจฉริยะแบบปิด
- 2) เพื่อศึกษาการทำงานของ Raspberry Pi
- 3) เพื่อศึกษาวิธีการส่งข้อมูลแบบไร้สาย
- 4) เพื่อศึกษาการทำงานระบบฐานข้อมูล
- 5) สร้างระบบที่สามารถตรวจสอบการทำงานภายในฟาร์มอัจฉริยะแบบปิดกับ เชิร์ฟเวอร์
- 6) สร้างระบบติดต่อระหว่างฟาร์มอัจฉริยะแบบปิดกับ เชิร์ฟเวอร์
- 7) สร้าง เว็บแอพพลิเคชัน เพื่อควบคุมการทำงานระยะไกล

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เกย์ตติกรสามารถดูควบคุมการทำงานผ่านเว็บแอพพลิเคชัน โดยไม่ต้องเดินทางไปในฟาร์ม
- 2) เกย์ตติกรสามารถตั้ง วัน เดือน ปี เวลา การทำงาน ได้อัตโนมัติ
- 3) สามารถนำไปประยุกต์ใช้กับฟาร์มแบบปิดที่มีอยู่ในปัจจุบัน

1.4 ขอบเขตของโครงการ

- 1) สร้างฟาร์มอัจฉริยะแบบปิดจำลอง
- 2) สร้างระบบบันทึกการทำงาน ภายในฟาร์มอัจฉริยะแบบปิด
- 3) สร้างระบบส่งข้อมูลจากฟาร์มอัจฉริยะแบบปิดไปยัง เชิร์ฟเวอร์
- 4) สร้าง เว็บแอพพลิเคชัน ควบคุมการทำงาน

1.5 ขั้นตอนการดำเนินงาน

- 1) เสนอแนวทางและเลือกเทคโนโลยีที่เหมาะสมกับโครงการ
- 2) ศึกษาเทคโนโลยีที่เกี่ยวข้อง
- 3) ศึกษาและออกแบบระบบโดยรวม
- 4) สร้างระบบย่อย ได้แก่
 - 4.1) แบบจำลองฟาร์มอัจฉริยะแบบปิด
 - 4.2) ระบบติดต่อระหว่างฟาร์มอัจฉริยะแบบปิดและเว็บแอพพลิเคชัน

บทที่ 2

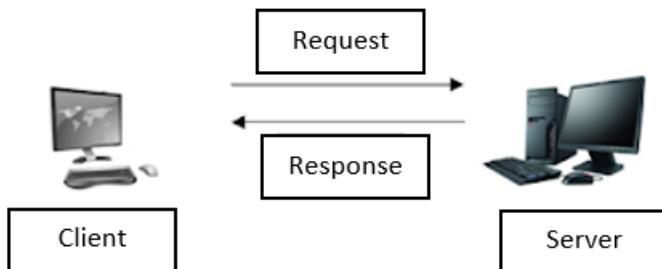
ทฤษฎีที่เกี่ยวข้อง

2.1 ความรู้เบื้องต้นการเขียนโปรแกรมบนอินเทอร์เน็ต

ลักษณะการเขียนโปรแกรมจะสามารถแบ่งได้เป็น 2 ประเภท คือ การเขียนโปรแกรมบนวินโดวัส (Windows-Based Application) และการเขียนโปรแกรมบนเว็บ (Web-Based Application) โดยการเขียนโปรแกรมบนเว็บลักษณะที่สำคัญก็คือ การแสดงผลของโปรแกรมจะอยู่บนบราวเซอร์ (Browser) ผู้ใช้สามารถเข้าถึงได้ผ่านทางเครื่องข่ายอินเทอร์เน็ต ไม่ว่าจะอยู่ที่ไหน หรือจะเข้าใช้งานเมื่อไรก็สามารถใช้งานได้ตลอดเวลา ซึ่งเป็นจุดเด่นที่ทำให้โปรแกรมบนเว็บมีเห็นอกว่าโปรแกรมบนวินโดวัส

2.1.1 หลักการทำงานของ WWW

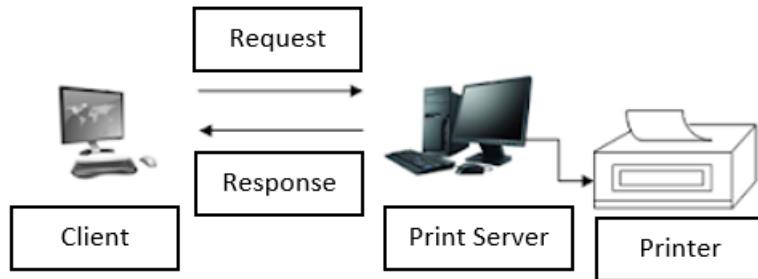
การทำงานของ World Wide Web หรือ WWW เป็นการทำงานในลักษณะไคลเอนต์ เชิร์ฟเวอร์ (Client-Server) คือ มีลักษณะของการเชื่อมต่อระหว่างเชิร์ฟเวอร์ซึ่งเป็นเครื่องผู้ให้บริการ และไคลเอนต์ซึ่งเครื่องผู้ใช้บริการ ดังรูปต่อไปนี้



รูปที่ 2.1 แสดงการทำงานของ Client-Server

จากรูป การทำงานจะเริ่มจากเครื่องไคลเอนต์ ทำการร้องขอบริการ (Request) จากเครื่องเชิร์ฟเวอร์ หลังจากเครื่องเชิร์ฟเวอร์ทำการจัดเตรียมข้อมูลหรือบริการตามที่เครื่องไคลเอนต์ได้ร้องขอมา ก็จะทำการตอบกลับ (Response) ไปยังเครื่องไคลเอนต์ โดยปกติเครื่องเชิร์ฟเวอร์จะมีอยู่เพียงเครื่องเดียว ในขณะที่เครื่องไคลเอนต์อาจมีได้หลายๆ เครื่อง และเครื่องไคลเอนต์หลายๆ เครื่องนี้ ก็อาจจะเข้ามาขอใช้บริการจากเครื่องเชิร์ฟเวอร์พร้อมกันได้

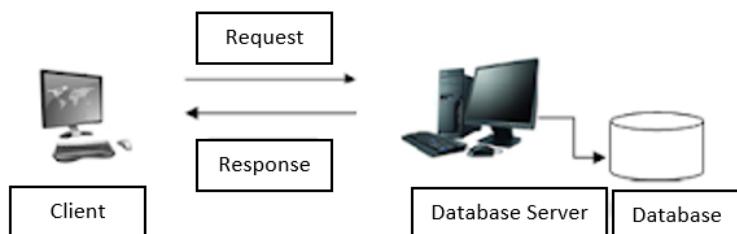
ในสำนักงานต่างๆ ได้นำประโภชน์ของการทำงานในลักษณะ ไคลเอนต์-เซิร์ฟเวอร์นี้ไปใช้จัดการงานต่างๆ ภายในสำนักงาน ดังรูปด้านล่าง



รูปที่ 2.2 แสดงการทำงานของไคลเอนต์-เซิร์ฟเวอร์ในลักษณะของการให้บริการงานพิมพ์

จากรูป เครื่องเซิร์ฟเวอร์ ได้ทำการเชื่อมต่อกับเครื่องพิมพ์ (Printer) ดังนั้นเครื่องเซิร์ฟเวอร์ นี้เราจะเรียกว่า ปรินต์เซิร์ฟเวอร์ (Print Server) ก็อเป็นเครื่องที่ให้บริการงานทางด้านการพิมพ์ กระบวนการการทำงานจะเริ่มต้นจากเครื่องไคลเอนต์ที่ต้องการพิมพ์เอกสารจึง ได้ทำการสั่งพิมพ์จาก เครื่องไคลเอนต์ดังกล่าว ข้อมูลที่ต้องการพิมพ์จะถูกส่งมาอยู่เครื่องปรินต์เซิร์ฟเวอร์ ซึ่งก็จะทำการ พิมพ์ข้อมูลออกทางเครื่องพิมพ์ ข้อดีของการเชื่อมต่อในลักษณะนี้ก็คือ ในสำนักงานนั้นก็เพียงแต่ จัดหาเครื่องพิมพ์ 1 เครื่องเชื่อมต่อกับเครื่องคอมพิวเตอร์ที่จะใช้เป็นปรินต์เซิร์ฟเวอร์ เมื่อพนักงาน ภายในบริษัทด้องการสั่งพิมพ์เอกสาร ใดๆ ก็สามารถสั่งงานที่เครื่องไคลเอนต์ได้เลย ประหยัด งบประมาณที่จะต้องทำการจัดหาเครื่องพิมพ์ให้กับพนักงานทุกๆ คน

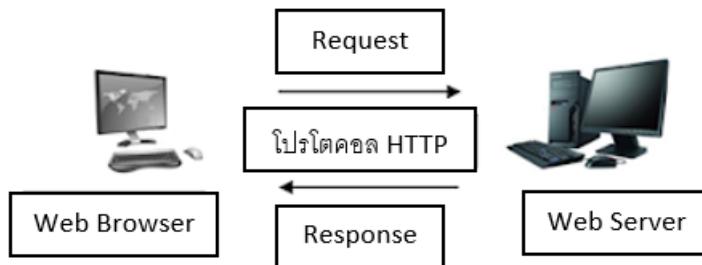
พิจารณาดัวอย่างของการทำงานแบบ ไคลเอนต์-เซิร์ฟเวอร์ อีกดัวอย่าง ดังรูปด้านล่าง



รูปที่ 2.3 แสดงการทำงานของไคลเอนต์-เซิร์ฟเวอร์ในลักษณะของการให้บริการฐานข้อมูล

จากรูปแสดงการทำงานของ ไคลเอนต์-เซิร์ฟเวอร์ เป็นอีกตัวอย่างของการทำงานในลักษณะ ไคลเอนต์-เซิร์ฟเวอร์ โดยเครื่องเซิร์ฟเวอร์ทำหน้าที่จัดเก็บข้อมูลต่างๆ ทั้งหมดของบริษัทไว้ในฐานข้อมูล ดังนั้นเครื่องเซิร์ฟเวอร์นี้เราจะเรียกว่า ค่าตัวเบสเซิร์ฟเวอร์ (Database Server) การทำงานก็จะเริ่มจากเครื่อง ไคลเอนต์ เช่นเดียวกัน คือเครื่อง ไคลเอนต์ อาจจะทำการเพิ่ม ลบ แก้ไข หรือ ดึงข้อมูลซึ่งจัดเก็บไว้ที่ฐานข้อมูล ข้อดีของการเชื่อมต่อแบบนี้คือ การจัดการข้อมูลต่างๆ ซึ่งถูกจัดเก็บไว้เพียงที่เดียวทำให้ข้อมูลมีความเป็นหนึ่งเดียว (Uniqueness) คือข้อมูลไม่กระชับกระจายและมีความถูกต้อง

จากที่ส่องตัวอย่างของการทำงานของ ไคลเอนต์-เซิร์ฟเวอร์ ลำดับถัดไปจะกล่าวถึงลักษณะของการทำงานของ WWW ซึ่งจะมีลักษณะการทำงานเหมือนกับ 2 ตัวอย่างที่ได้อธิบายไปแล้วพิจารณาปัจจุบันนี้



รูปที่ 2.4 แสดงการทำงานของ WWW

จากรูป เครื่องเซิร์ฟเวอร์ในที่นี้จะให้บริการต่างๆ ที่เกี่ยวกับเว็บทั้งหมด ไฟล์เว็บเพจ รูปภาพ หรือโปรแกรมบนเว็บต่างๆ จะถูกจัดเก็บไว้ในเครื่องนี้ ซึ่งเราจะเรียกเครื่องนี้ว่า เว็บเซิร์ฟเวอร์ (Web Server) เมื่อเครื่อง ไคลเอนต์ ขอใช้บริการ เว็บไซต์ ก็จะถูกแสดงบนเว็บбраузอร์ (Web Browser) เช่น Internet Explorer หรือ Firefox เป็นต้น กระบวนการทำงานจะเริ่มจากเว็บбраузอร์จะทำการร้องขอหน้าเว็บใดๆ โดยการพิมพ์ URL (Universal Resource Locator) จากโปรแกรมбраузอร์ซึ่งข้อมูลจะถูกกระทำผ่าน โปรโตคอล HTTP (Hypertext Transfer Protocol) เครื่องเว็บเซิร์ฟเวอร์เมื่อได้รับคำร้องขอ ก็จะทำการจัดส่งหน้าเว็บเพจนั้นผ่านไปทางเว็บбраузอร์

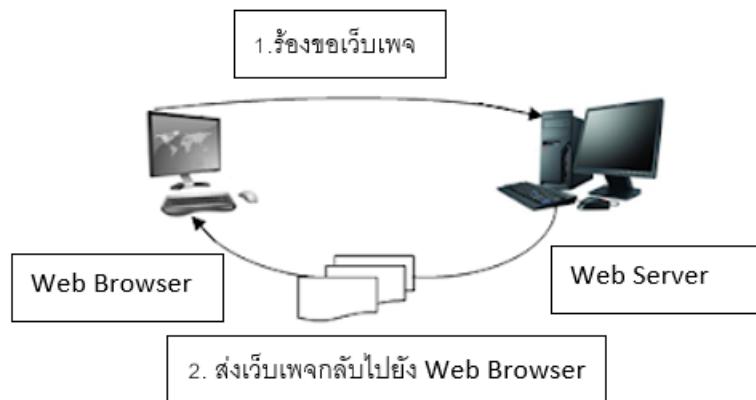
2.1.2 ประเภทของโปรแกรมบนเว็บ

ในการเขียนโปรแกรมบนเว็บสามารถที่จะแบ่งลักษณะการทำงานของโปรแกรมได้เป็น 2 ประเภท คือ

1) Static Programming

2) Dynamic Programming

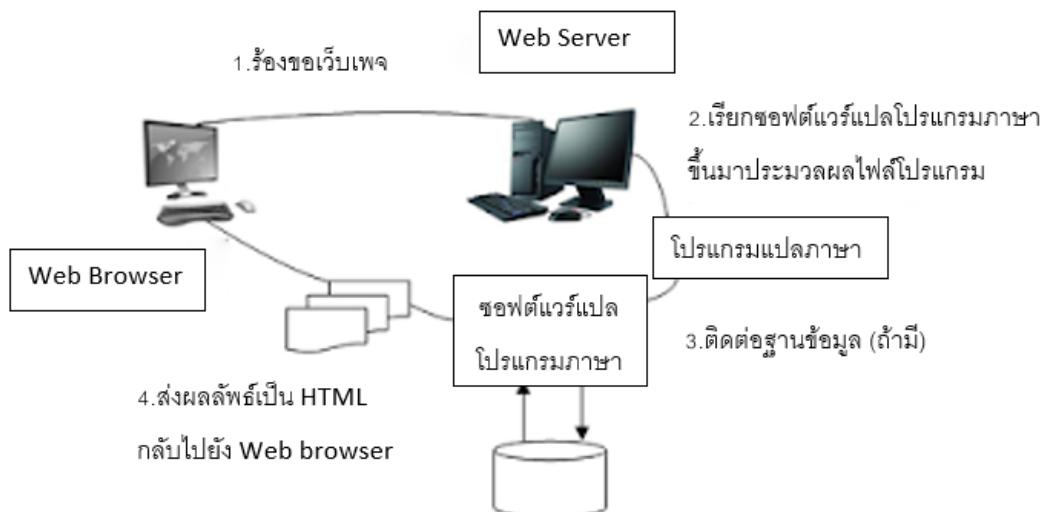
Static Programming เป็นลักษณะของโปรแกรมบนเว็บที่ไม่ค่อยมีการเปลี่ยนแปลง เช่น เว็บประวัติส่วนตัว, เว็บนำเสนอประวัติและโครงสร้างขององค์กรนั้น เป็นต้น ลักษณะเว็บประเภทนี้ เมื่อผู้พัฒนาเว็บสร้างขึ้นมาแล้วหากต้องการที่จะทำการแก้ไขข้อมูลบางอย่างนั้นก็จะต้องใช้ โปรแกรมในการสร้างเว็บเพจ เช่น Adobe Dreamweaver, Microsoft FrontPage เป็นต้น เปิดไฟล์ของหน้าเว็บนั้นแล้วรีบทำการแก้ไขข้อมูลที่ต้องการ เมื่อเสร็จแล้วก็ต้องทำการบันทึกไฟล์ดังกล่าว และทำการอัปโหลดไฟล์ขึ้นไปเก็บไว้ที่เซิร์ฟเวอร์จะเห็นว่าลักษณะของโปรแกรมบนเว็บประเภทนี้ จะไม่มีความหยุดหย่อนในการจัดการ และสร้างความยุ่งยากให้แก่ผู้พัฒนาเว็บเพจด้วย ลักษณะการทำงานของโปรแกรมบนเว็บประเภทนี้มีลักษณะดังรูปด้านล่าง



รูปที่ 2.5 แสดงการทำงานของ Static Programming

Dynamic Programming เป็นลักษณะของโปรแกรมบนเว็บที่เกิดขึ้นมาเพื่อแก้ปัญหาความไม่หยุดหย่อนในการจัดการข้อมูลของ Static Programming โดยหมายความว่าระบบเว็บที่ต้องการเปลี่ยนแปลงข้อมูลบ่อยครั้งหรือเว็บที่มีการจัดเก็บข้อมูลจำนวนมากไว้ภายในฐานข้อมูล เช่น เว็บหนังสือพิมพ์, เว็บแสดงรายละเอียดของสินค้า เป็นต้น

ในการพัฒนาโปรแกรมบริเว็บประเภทนี้จะต้องอาศัยผู้พัฒนาโปรแกรมที่มีความรู้ความสามารถในการเขียนโปรแกรม นอกจานี้ในส่วนของซอฟต์แวร์ที่ต้องติดตั้งก็จะประกอบไปด้วยซอฟต์แวร์ที่ใช้ในการทำให้เครื่องคอมพิวเตอร์มีความสามารถเป็นเครื่องเว็บเซิร์ฟเวอร์, ซอฟต์แวร์ที่ใช้ในการแปลงโปรแกรมภาษาและฐานข้อมูล พิจารณาลักษณะการทำงานของโปรแกรมบนเว็บประเภทนี้ ดังรูปต่อไปนี้



รูปที่ 2.6 แสดงการทำงานของ Dynamic Programming

จากรูป โปรแกรมก็จะเริ่มกระบวนการทำงานจากเครื่องไคลเอนต์ทำการร้องขอเว็บเพจ ซึ่งเว็บเพจที่ร้องขอได้มีการเขียนโปรแกรมบนเว็บในลักษณะ Dynamic Programming ได้ ดังนี้น เครื่องเซิร์ฟเวอร์ก็จะทำการเรียกซอฟต์แวร์แปลงโปรแกรมภาษาชื่นมาเพื่อแปลงโปรแกรมภาษา (Programming Language) ให้เป็นภาษาเครื่อง (Machine Language) เนื่องจากเครื่องคอมพิวเตอร์จะเข้าใจเฉพาะภาษาเครื่องเท่านั้น ถ้าโปรแกรมที่เขียนขึ้นมีคำสั่งที่ทำการจัดการฐานข้อมูลก็จะทำการเพิ่ม ลบ แก้ไข หรือดึงข้อมูลจากฐานข้อมูล ซึ่งโดยปกติตัวฐานข้อมูลนี้อาจจะอยู่ภายในเครื่องเซิร์ฟเวอร์ หรืออาจจะแยกออกมานเป็นเครื่องคอมพิวเตอร์อีกเครื่องก็ได้ หลังจากนั้นขั้นตอนสุดท้าย หลังจากเครื่องเซิร์ฟเวอร์ทำการประมวลผลเสร็จเรียบร้อยแล้วก็จะทำการส่งข้อมูลผลลัพธ์ในลักษณะ HTML กลับไปยังเครื่องไคลเอนต์ หรือเครื่องที่ทำการร้องขอหนึ่ง

2.2 การติดตั้งโปรแกรม XAMPP

ในการพัฒนาโปรแกรมภาษา PHP จะต้องติดตั้งโปรแกรม ได้แก่ PHP Engine, Apache Web Server, MySQL Database และ PhpMyAdmin แต่การจะติดตั้งใช้งานที่ละ โปรแกรมจะเสียเวลามาก และยังต้องมีการปรับแต่งค่าต่างๆ ในแต่ละ โปรแกรมอีกด้วย จึงเกิด โปรแกรม XAMPP ขึ้นมา ซึ่ง โปรแกรม XAMPP คือชุดการติดตั้งเพียงครั้งเดียว และว่าได้ โปรแกรมทั้งหมดพร้อมกันเลย และไม่ ต้องเสียเวลาปรับแต่งค่าของ โปรแกรมให้ยุ่งยากอีกด้วย

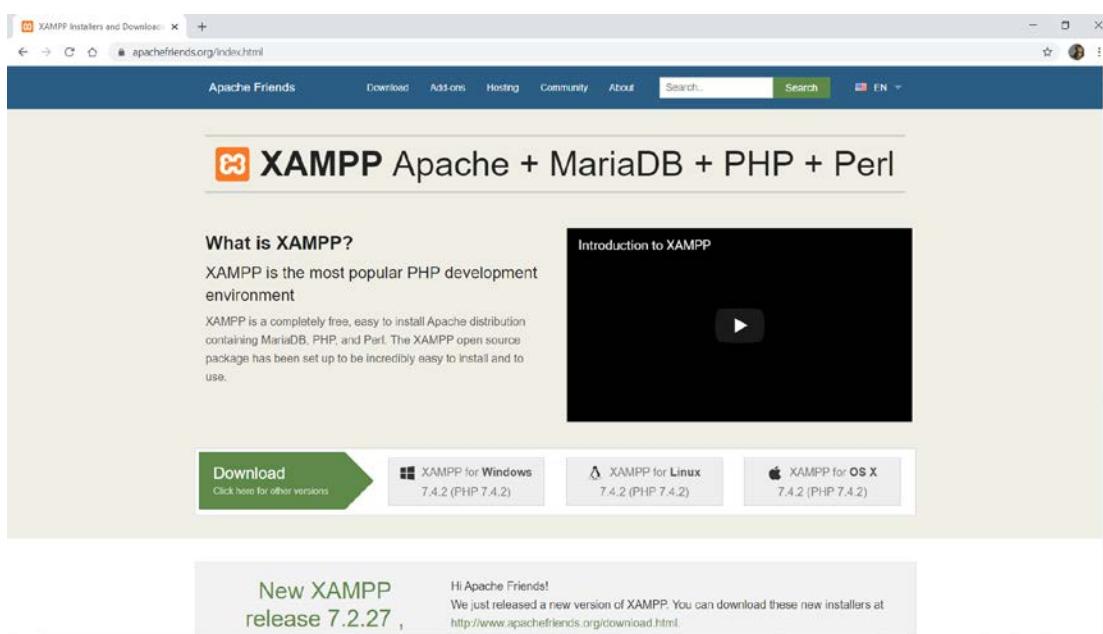
โปรแกรม XAMPP ที่จะติดตั้งนี้ประกอบไปด้วย

- Apache Web Server
- PHP Script Language
- MySQL Database
- PhpMyAdmin Database Manager

2.2.1 ขั้นตอนการติดตั้งโปรแกรม XAMPP

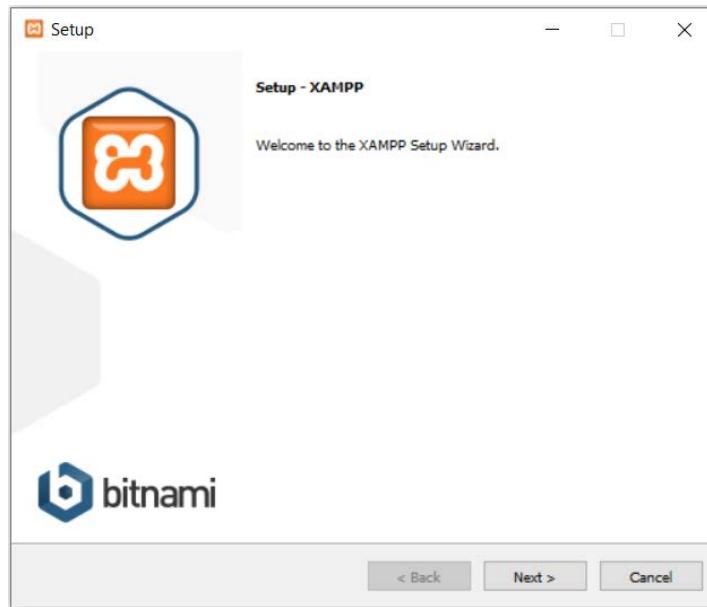
1. เริ่มต้นให้ไปดาวน์โหลด โปรแกรม XAMPP ได้ที่

<https://www.apachefriends.org/download.html> จากนั้นให้ติดตั้งตามขั้นตอน ดังนี้



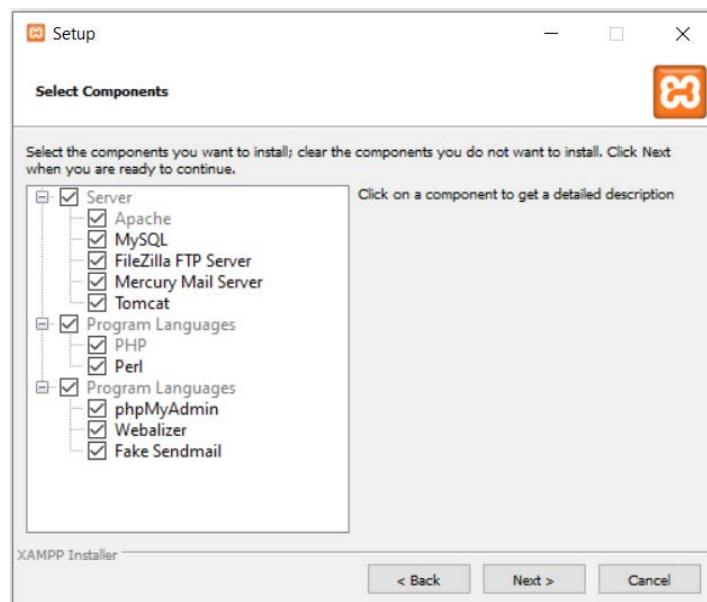
รูปที่ 2.7 แสดงหน้าเว็บ XAMPP Apache

2. เข้าสู่หน้าจอ Welcome ของโปรแกรมให้ทำการคลิกปุ่ม Next >



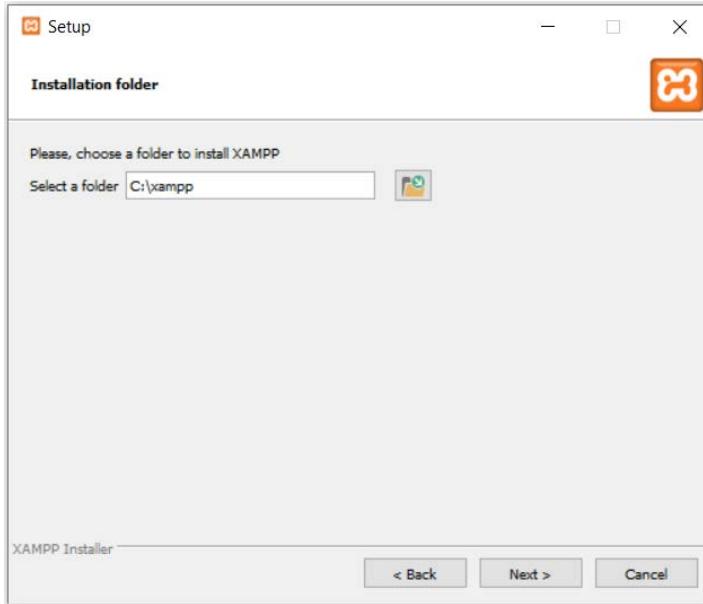
รูปที่ 2.8 เริ่มต้นการติดตั้งโปรแกรม XAMPP

3. เข้าสู่หน้าจอของการเลือกคอมโพเนนต์ที่ต้องการติดตั้ง ในกรณีนี้ให้เลือกทุกตัว เสร็จแล้วทำการคลิกที่ปุ่ม Next >



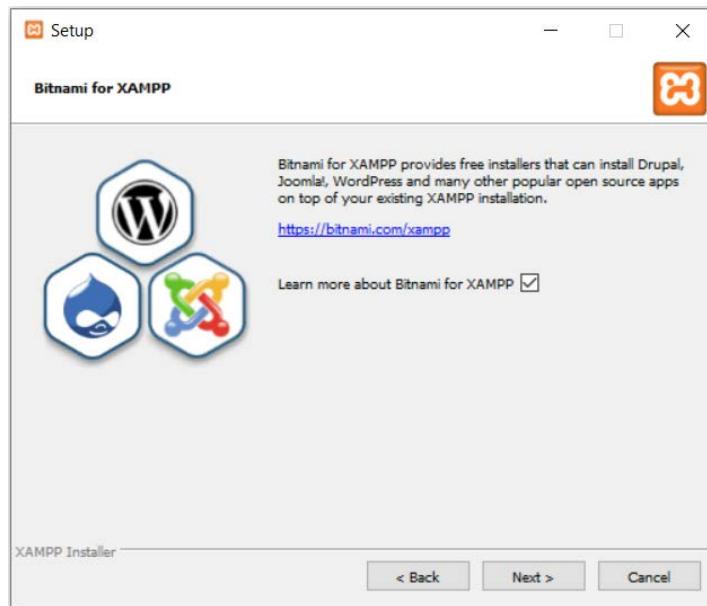
รูปที่ 2.9 หน้าจอการเลือก Components ที่ต้องการติดตั้ง

4. เข้าสู่หน้าของการเลือกไดรฟ์และโฟลเดอร์ที่ต้องการติดตั้ง ซึ่งจะ default ไว้ที่ C:\xampp ในที่นี่จะเลือกเป็นไดรฟ์และโฟลเดอร์สำหรับการติดตั้งให้ทำการคลิกที่ปุ่ม Next >



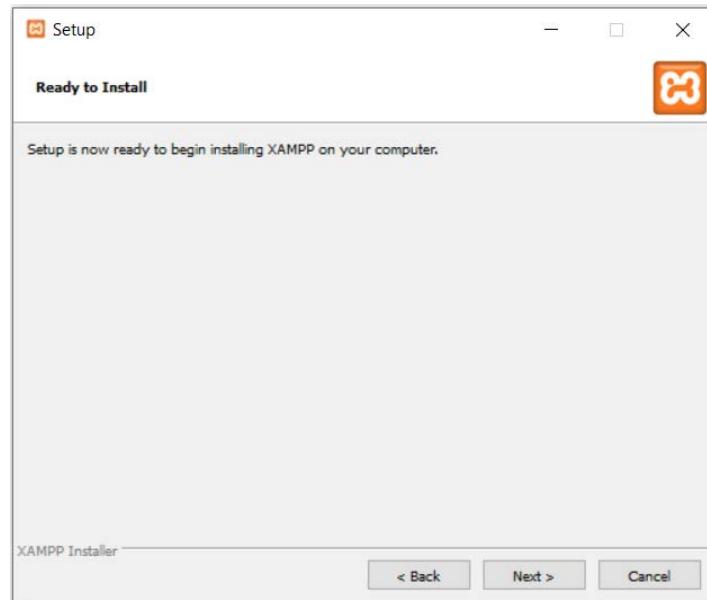
รูปที่ 2.10 หน้าของการกำหนดไดรฟ์และโฟลเดอร์สำหรับการติดตั้ง

5. เนื่องจากโปรแกรม XAMPP จะสนับสนุนการใช้งานโปรแกรม Drupal, Joomla และ World-press จำต้องการติดตั้งเพิ่มเติมให้เสร็จแล้วให้ทำการคลิกที่ปุ่ม Next >



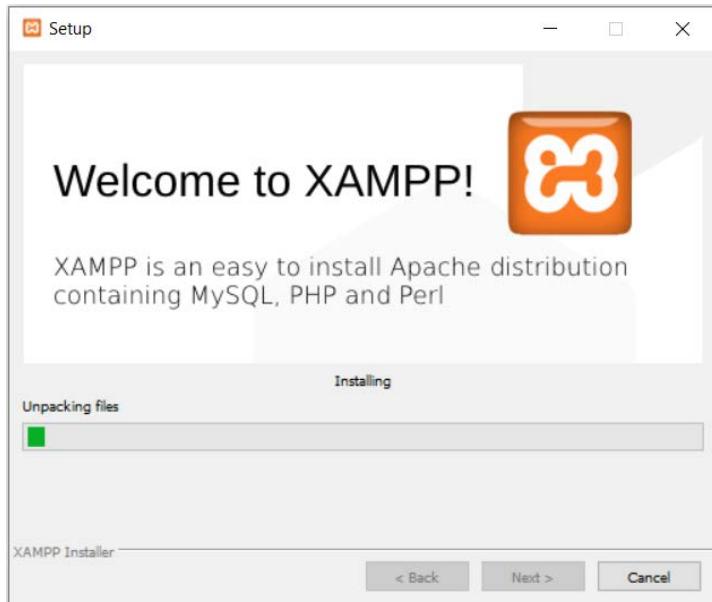
รูปที่ 2.11 หน้าจอแสดงการตระเตรียมโปรแกรม Drupal, Joomla, World-press

6. ในส่วนของหน้าจอของการติดตั้งจะเข้าสู่ขั้นตอนของความพร้อมก่อนการติดตั้งให้ทำการคลิกที่ปุ่ม Next >



รูปที่ 2.12 หน้าจอ ก่อนเริ่มทำการติดตั้ง

7. จะเริ่มทำการติดตั้งโปรแกรม XAMPP ให้รอจนกว่าการติดตั้งโปรแกรมเสร็จสมบูรณ์



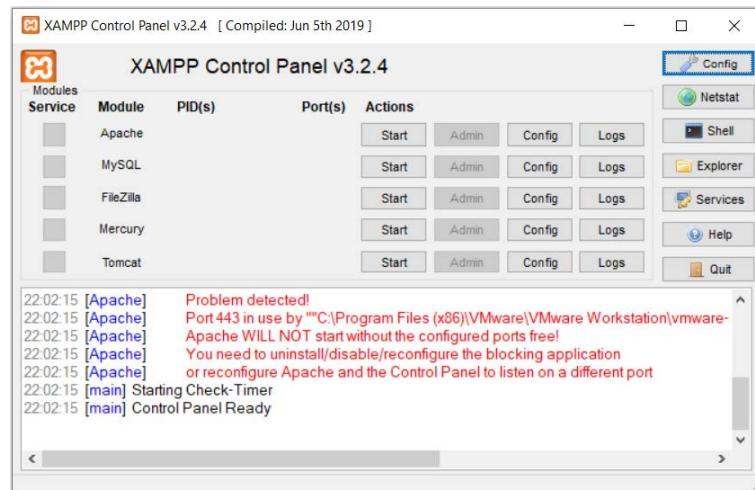
รูปที่ 2.13 หน้าจอแสดงระหว่างทำการติดตั้ง

8. หลังจากทำการติดตั้งโปรแกรม XAMPP เสร็จสมบูรณ์ ให้ทำการคลิกที่ปุ่ม Finish ซึ่งในกรณีนี้เราเลือกที่จะให้เริ่มการทำงานของ Control Panel ด้วย (เลือกที่รายการ Do you want to start the control Panel now?)



รูปที่ 2.14 หน้าจอแสดงการติดตั้งที่เสร็จสมบูรณ์

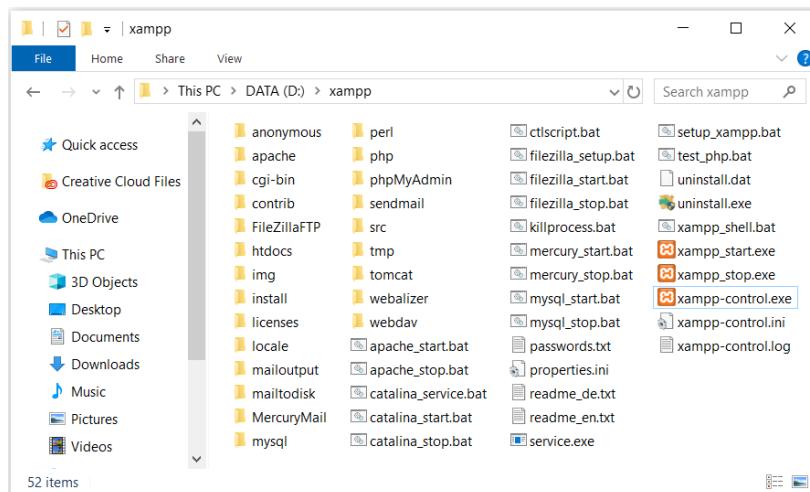
9. จะเปิดโปรแกรม Control Panel ขึ้นมา ซึ่งจะเป็นส่วนที่ใช้สำหรับการควบคุมการทำงานของโปรแกรมทั้งหมด ให้สังเกตที่ชื่อโปรแกรมที่เราต้องการใช้งานว่าสถานะทำงานอยู่หรือไม่ (Action) ถ้าไม่ทำงานให้ทำการคลิกที่ปุ่ม Start ของโปรแกรมดังกล่าว



รูปที่ 2.15 หน้าจอ Control Panel

2.2.2 ระบบการจัดเก็บไฟล์ของโปรแกรม XAMPP

หลังจากติดตั้งโปรแกรม XAMPP เสร็จเรียบร้อยแล้ว โปรแกรมจะจัดเก็บไฟล์ต่างๆ ทั้งหมดไว้ในไดร์ฟที่ติดตั้งโฟลเดอร์ xampp



รูปที่ 2.16 แสดงโฟลเดอร์และไฟล์ที่จัดเก็บ

รายละเอียดของแต่ละ โฟลเดอร์ภายใน xampp ที่ควรรู้ก็มีรายละเอียด ดังนี้

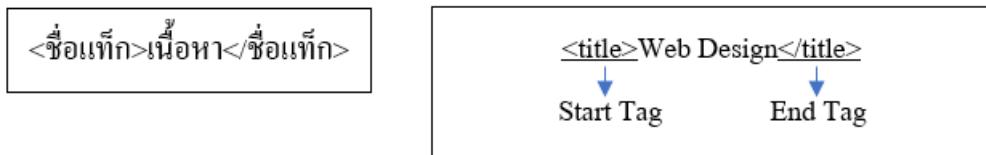
- apache\conf\httpd.conf
เป็นไฟล์สำหรับปรับแต่ง Apache Web Server
- htdocs\
เป็นที่เว็บไซต์และไฟล์ เช่น ไฟล์ PHP
- mysql\data\
เป็นที่เก็บฐานข้อมูล MySQL
- phpMyAdmin\libraries\config.default.php
เป็นไฟล์สำหรับเปลี่ยน user และ password ในการเชื่อมต่อ MySQL
- php\php.ini
เป็นไฟล์สำหรับปรับแต่งภาษา PHP
- mysql\bin\my.ini
เป็นไฟล์สำหรับปรับแต่งฐานข้อมูล MySQL
- mysql\bin\
เป็นที่เก็บคำสั่ง MySQL
- apache\bin\
เป็นที่เก็บคำสั่ง Apache

2.3 HTML Basic Element & Format

HTML (Hypertext Markup Language) เป็นภาษาในการกำหนดโครงสร้างของเอกสารเว็บเพจ โดย HTML นั้นไม่ใช่ว่าเป็นภาษาในการเขียนโปรแกรม (Programming Language) แต่ใช้เป็นภาษาประเภทการกำหนดสัญญาลักษณ์ (Markup Language) ในรูปแบบแท็ก (Tag) เช่น <html>, <head>, <body> เป็นต้น ถึงแม้ในปัจจุบันมีเทคโนโลยีเกี่ยวกับเว็บออกแบบอย่างมากมาย แต่เราใช้ HTML ในการจัดวางโครงสร้างของเอกสารโดยไม่มีสิ่งใดทดแทนได้ สำหรับในโครงการนี้เราจะได้ศึกษาเกี่ยวกับแท็กพื้นฐานที่สำคัญของ HTML สำหรับการกำหนดและจัดรูปแบบของข้อความ

2.3.1 ลักษณะของแท็กและอิเลิมต์

องค์ประกอบต่างๆ ภายในเว็บเพจ จะถูกกำหนดด้วยสัญลักษณ์ที่เรียกว่า แท็ก (Tag) เพื่อแสดงจุดเริ่มต้นและจุดสิ้นสุด (หรือขอบเขต) ของเนื้อหาในแต่ละอย่าง ดังรูปแบบต่อไปนี้



รูปที่ 2.17 ลักษณะของแท็ก

เราสามารถสรุปลักษณะพื้นฐานที่สำคัญเกี่ยวกับแท็กของ HTML ได้ดังนี้

- ชื่อแท็ก จะกำหนดระหว่างเครื่องหมาย < และ > เช่น <html> หรือ <title> เป็นต้น
- แท็กที่กำหนดไว้ที่จุดเริ่มต้นของเนื้อหาจะเรียกว่า Start Tag
- แท็กที่กำหนดไว้ที่จุดสิ้นสุดของเนื้อหาจะเรียกว่า End Tag และเขียนเครื่องหมาย / ไว้หน้าชื่อแท็กด้วย เช่น </html> เป็นต้น
- เมื่อร่วมแท็กและเนื้อหาเข้าด้วยกัน เราจะเรียกว่า อิลิเม้นต์ (Element) ดังนั้น จึงกล่าวได้ว่า อิลิเม้นต์ ก็คือสิ่งที่อยู่ระหว่าง Start Tag และ End Tag นั้นเอง
- ถ้าแท็กเริ่มต้นและสิ้นสุดต้องเป็นชื่อเดียวกัน เช่น <body> กับ </body> หรือ <title> กับ </title> เป็นต้น
- เราสามารถเขียนคู่แท็กเริ่มต้นและสิ้นสุดด้วยรูปแบบตัวพิมพ์ที่ต่างกันได้ เช่น <title> กับ </TITLE> เป็นต้น แต่โดยทั่วไปนิยมเขียนแท็กด้วยตัวพิมพ์แบบเดียวกันทั้งหมด

- ยังมีอิลิเม้นต์อีกกลุ่มหนึ่งที่ไม่ต้องกำหนดเนื้อหาให้กับมัน โดยจะเขียนด้วยแท็กเดี่ยว รูปแบบเดี่ยว เช่น `
` เป็นต้น
- เนื้อหาของอิลิเม้นต์อาจจะเป็นข้อความธรรมชาติ หรือนำอิลิเม้นต์อื่นๆ มาวางช้อนกันก็ได้ เช่น

```
<head>
    <title>Web Design</title>
</head>
```

2.3.2 แอตทริบิวต์ของแท็ก

แอตทริบิวต์ (Attribute) เป็นการกำหนดข้อมูลบางอย่างให้กับแท็กต่างๆ เช่น แท็กที่ใช้แสดงรูปภาพ ก็จะมีแอตทริบิวต์ที่ใช้กำหนดตำแหน่งภาพ หรือแท็กที่แสดงลิงก์ ก็จะมีแอตทริบิวต์ที่ใช้กำหนด URL ปลายทางเมื่อคลิกที่ลิงก์ เป็นต้น การกำหนดแอตทริบิวต์มี 2 ลักษณะดังนี้

2.3.2.1 แอตทริบิวต์แบบ name=value

แอตทริบิวต์แบบนี้จะกำหนดชื่อและค่าของมันในรูปแบบ `name="value"` ที่แท็กเริ่มต้นในรูปแบบ

```
<ชื่อแท็ก attribute1="value" ชื่อแท็ก attribute2="value" ชื่อแท็ก attribute3="value" ...>
```

แนวทางการกำหนดแอตทริบิวต์แบบ Name & Value เช่น

```
<p>HTML : Hypertext Markup Language

<a href="http://www.reg.kmitl.ac.th" target="_blank">GO</a>
```

สิ่งที่ควรรู้เพิ่มเติมเกี่ยวกับแอตทริบิวต์ มีดังนี้

- กำหนดค่าของแอตทริบิวต์นั้น สามารถทำได้ 3 ลักษณะคือ
 - ใช้เครื่องหมาย “ Double Quote เช่น ``
 - ใช้เครื่องหมาย ‘ Single Quote เช่น ``
 - ไม่ใช้เครื่องหมายใดเลย เช่น ``
- โดยทั่วไปเรานิยมกำหนดค่าโดยใช้เครื่องหมาย “มากกว่า ส่วนเครื่องหมาย ‘ ไม่นิยมใช้กันมากนัก ยกเว้นในบางสถานการณ์ เช่น กรณีที่ค่าของแอตทริบิวต์นั้นมีเครื่องหมาย “ รวมอยู่ด้วย เรายังต้องเลี่ยงไปใช้เครื่องหมาย ‘ กำหนดค่าแทนเช่น ``

- การกำหนดค่าแอตทริบิวต์แบบไม่มีเครื่องหมายใดๆ กำกับ จะใช้เฉพาะกรณีที่ค่าของแอตทริบิวต์นั้นเขียนติดกันโดยไม่มีช่องว่างหรืออักขระพิเศษบางตัว รวมอยู่ด้วยกันด้วย ถ้ามีช่องว่างกัน เบรเซอร์ตีความหมายว่า เลพาะคำแรกที่อยู่ก่อนช่องว่างเท่านั้นเป็นค่าของแอตทริบิวต์ เช่น หากเขียนเป็น เว็บเบรเซอร์จะมองว่า เลพาะ “my” เท่านั้นที่เป็นค่าของแอตทริบิวต์ alt ซึ่งอาจเกิดข้อผิดพลาดในบางสถานการณ์
- การเขียนชื่อของแอตทริบิวต์นั้น เราอาจเขียนด้วยตัวพิมพ์เล็กหรือใหญ่ หรือปะปนกันก็ได้ เช่น อาจเขียนเป็น หรือ ก็ได้แต่โดยทั่วไปเรา尼ยมเขียนด้วยตัวพิมพ์เล็กทั้งหมด

2.3.2.2 แอตทริบิวต์แบบบูลีน

แอตทริบิวต์แบบบูลีน (Boolean) จะกำหนดค่าในลักษณะ true/false เช่น แท็ก <input> จะมีแอตทริบิวต์ชื่อ disabled สำหรับการกำหนดสถานะของอินพุทว่าสามารถใช้งานได้ หรือไม่ ซึ่งอาจกำหนดค่าได้หลายลักษณะดังนี้

- กำหนดเพียงแค่ชื่อแอตทริบิวต์ โดยไม่กำหนดค่า เช่น

```
<input type="text" disabled>
```

- กำหนดค่าโดยนำชื่อแอตทริบิวต์มาระบุเป็นค่า เช่น

```
<input type="text" disabled="disabled">
```

- นอกจากนี้ เรายังเห็นการกำหนดค่าในลักษณะอื่นๆ อีก เช่น

```
<input type="text" disabled="">
```

```
<input type="text" disabled="true">
```

2.3.3 โครงสร้างหลักของ HTML

ลักษณะโครงสร้างพื้นฐานของเว็บเพจ สามารถเปลี่ยนในรูปแบบโค้ดของ HTML ได้ดังนี้

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
.....
```

```
</head>
```

```
<body>
```

```
.....
```

```
</body>
```

```
</html>
```

วัตถุประสงค์แต่ละแท็ก คือ

<!doctype html>	ประกาศชนิดของเอกสาร	<body>	ส่วนเนื้อหาของเอกสาร
<html>	เริ่มต้นเอกสาร	</html>	สิ้นสุดเอกสาร HTML
<head>	ส่วนก		

2.3.4 แท็กสำหรับหัวข้อและเอกสาร

ส่วนหัวของเว็บเพจ จะกำหนดด้วยแท็ก <head> แต่เราจะไม่ระบุแอ็ตทริบิวต์ให้กับมัน โดยตรง เพราะมีแท็กที่ใช้งานเฉพาะในแต่ละอย่างอยู่แล้ว ซึ่งแท็กที่เราควรรู้จักในเบื้องต้นมีดังนี้

แท็ก <title>

แท็ก <title> ใช้ในการกำหนดชื่อของเว็บเพจนั้น โดยวางชื่อนี้ไว้ภายในแท็ก <head> ดัง โค้ดด้านไป ซึ่งข้อมูลนี้จะถูกใช้โดย Search Engine ในการแสดงผลลัพธ์ของ Search Engine ด้วย รวมถึงการเพิ่มเพจใน Bookmark หรือการแสดงผลในผลลัพธ์ของ Search Engine ด้วย

```
<head>
    <title>Web Design</title>
</head>
```

แท็ก <meta>

แท็ก <meta> ใช้ในการกำหนดข้อมูลบางอย่างเกี่ยวกับเพจนั้น เช่น รหัสอักขระ คีย์เวิร์ด สำหรับ Search แต่ไม่มี End Tag โดยจะกำหนดค่าในแบบแอ็ตทริบิวต์ทั้งหมด เช่น

```
<head>
    <meta charset="utf-8">
</head>
```

แอ็ตทริบิวต์ของแท็ก <meta> แบ่งเป็นหลายลักษณะ สำหรับในที่นี้จะกล่าวเฉพาะสิ่งที่เรา นิยมใช้กันเป็นส่วนใหญ่ท่านนั้นคือ

Charset	ใช้กำหนดรหัสอักขระสำหรับเพจนั้น เช่น <meta charset="utf-8">
Content	ใช้ในการกำหนดค่าสำหรับแอ็ตทริบิวต์อื่น
http-equiv="refresh"	ให้เว็บเบราว์เซอร์โหลดเพจนั้นมาแสดงใหม่ทุกช่วงเวลาที่กำหนด (วินาที) โดยกำหนดเวลาด้วยแอ็ตทริบิวต์ content เช่น หากต้องการให้โหลดเพจ ทุก 10 นาที ซึ่งเท่ากับ 600 วินาที ก็กำหนดแท็กดังนี้ <meta http-equiv="refresh" content="600">

Name="description"	ใช้กำหนดคำอธิบายโดยสังเขปเกี่ยวกับเพจนั้น ซึ่งบางที่ Search Engine อาจจะนำไปใช้ในการแสดงผลลัพธ์จากการสืบค้นอีกด้วย โดยให้เขียนคำอธิบายด้วยแท็คทริบิวต์ content เช่น
Name="keyword"	กำหนดคำสำคัญหรือคีย์เวิร์ดที่มีในเพจนั้น เพื่อให้ Search Engine นำไปทำเป็นตัวชี้นำการสืบค้นข้อมูล โดยกำหนดคีย์เวิร์ดด้วยแท็คทริบิวต์ content หากมีมากกว่า 1 คำ ให้กันแต่ละคำด้วยเครื่องหมาย , เช่น <meta name="keyword" content="PHP,HTML,CSS,Bootstrap">

กรณีที่เราต้องการกำหนด Meta หลายอย่าง ให้เขียนแยกไว้คนละแท็ก เช่น

```
<head>
    <title>Web Design</title>
    <meta charset="utf-8">
    <meta http-equiv="refresh" content="600">
    <meta name="keyword" content="PHP,HTML,CSS,Bootstrap">
</head>
```

แท็ก <base>

แท็ก <base> ใช้ในการกำหนด URL สำเร็จ เพื่อหลีกเลี่ยงการที่จะต้องกำหนด URL แบบเต็ม เช่น สมมุติว่าโดยปกติเราต้องกำหนดตำแหน่ง URL ของหลายๆ ภาพซึ่งกันในลักษณะดังนี้

```


```

เราสามารถนำ URL ส่วนที่ต้องการอ้างถึงซ้ำๆ กัน ไปกำหนดด้วยแท็ก <base> ไว้ที่ส่วนหัว (head) ของเพจแล้วต่อไปก็สามารถอ้าง URL เลพางส่วนที่อยู่ต่อจาก base ได้เลย แต่ URL ที่ไม่ได้อยู่ที่เดียวกับ base ต้องระบุแบบเต็มๆ ตามปกติ เช่น

ลักษณะที่เกี่ยวกับแท็ก <base> ที่ควรรู้จักเพิ่มเติม มีดังนี้

- ในแต่ละเพจ จะมีแท็ก <base> ได้เพียงอันเดียวเท่านั้น

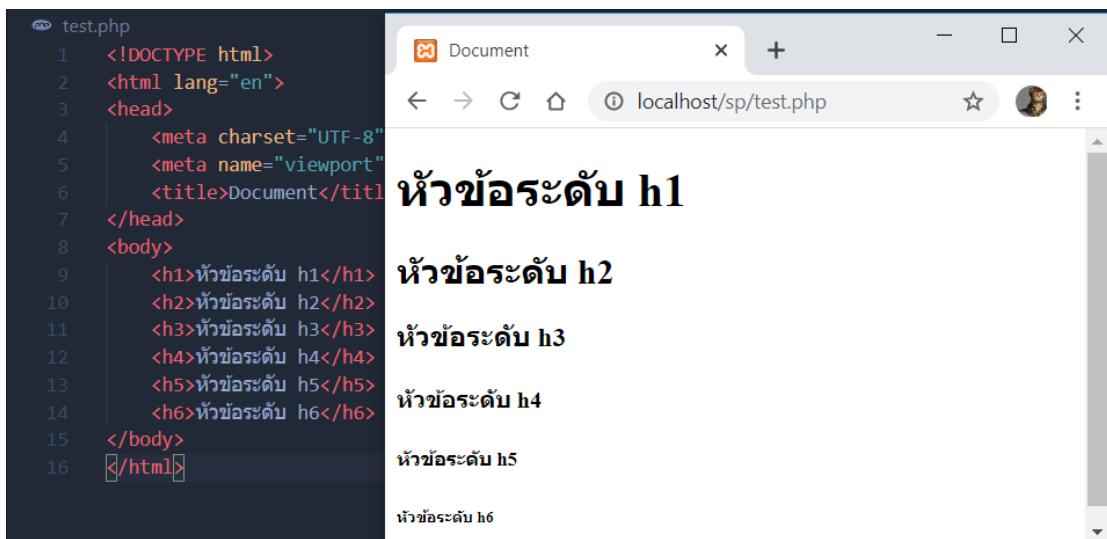
```
<head>
    <base href="http://www.developerthai.com/img/">
</head>
<body>
    
    
</body>
```

- แท็ก `<base>` ยังมีอีกหนึ่งแอตทริบิวต์คือ `target` ซึ่งมักใช้ในกรณีที่เราต้องการให้เปิดเพจเป้าหมายบนแท็บใหม่ โดยกำหนดค่าแอตทริบิวต์นี้เป็น `“_blank”` เช่น `<base href="http://www.example.com" target="_blank">`

2.3.5 แท็กสำหรับกำหนดขนาดหัวข้อ

เรามักจะทำให้ข้อความที่เป็นหัวข้อ ดูโดดเด่นกว่าส่วนที่เป็นรายละเอียด ดังนั้น HTML จึงมีแท็กที่ใช้ในการกำหนดขนาดข้อความที่เป็นหัวข้อให้เลือกใช้ได้ถึง 6 ระดับ คือ

<code><h1>, <h2>, <h3>, ..., <h6></code>	กำหนดขนาดหัวข้อที่ต่างกัน 6 ระดับ โดย <code><h1></code> มีขนาดใหญ่ที่สุด และ <code><h2>, <h3>, ..., <h6></code> จะมีขนาดลดหลั่นกันลงมาตามลำดับ
--	--



รูปที่ 2.18 แนวทางการกำหนดหัวข้อด้วย `<hx>`

2.3.6 แท็กสำหรับการแบ่งส่วนของข้อความ

ในกรณีที่เราต้องการแยกข้อความบางส่วนให้เป็นอีกย่อหน้าหนึ่ง หรือการแบ่งกลุ่มข้อความเพื่อการจัดรูปแบบเฉพาะ ก็สามารถกำหนดข้อความดังกล่าวไว้ในแท็กต่อไปนี้

<code><p></code>	สำหรับการเขียนย่อหน้าใหม่ โดยกำหนดข้อความไว้ระหว่างแท็กเปิด <code><p></code> และแท็กปิด <code></p></code> แล้วข้อความนั้นจะถูกนำไปขึ้นบรรทัดใหม่ โดยเว้นระยะห่างพอประมาณ
<code><div></code>	จะใช้จัดกลุ่มเนื้อหาแบบ (Block) ซึ่งตามปกตินั้น เนื้อหาที่อยู่ในอลิเมนต์ <code>div</code> ไม่สามารถอ่านอยู่ในบรรทัดเดียวกันกับเนื้อหาที่อยู่นอก <code>div</code> ได้ แต่จะถูกแยกออกไปอยู่บรรทัดใหม่ต่างหาก โดยเนื้อหาที่มีใส่ใน <code>div</code> ไม่จำเป็นต้องเป็นข้อความอย่างเดียวเท่านั้น แต่สามารถเป็นอลิเมนต์ประเภทอื่นๆ ได้ทั้งหมด เช่น รูปภาพ ฟอร์ม ตาราง เป็นต้น ซึ่งส่วนใหญ่เรามักใช้ร่วมกับการจัดรูปแบบด้วย CSS

	จะใช้จัดกลุ่มนิ้อหางายในบรรทัด (Inline) โดยเนื้อหาที่อยู่ในอิลิเม้นต์ span สามารถรวมอยู่ในบรรทัดเดียวกันกับเนื้อหาที่อยู่นอก span ได้
 	ใช้ในการกำหนดจุดสุดบรรทัด (Line Break) เพื่อขึ้นบรรทัดใหม่
<hr>	สำหรับแสดงเส้นแบ่งในแนวนอน (Horizontal Rule)

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="w
6   <title>Document</title>
7 </head>
8 <body>
9   <p>1 2</p><p>3 4</p>
10  <hr>
11  <div>1 2</div><div>3 4</div>
12  <span>5 6</span><span>7 8</span>
13  <br><br>
14  <span>9 10</span><br>
15  <span>11 12</span>
16  <hr>
17 </body>
18 </html>

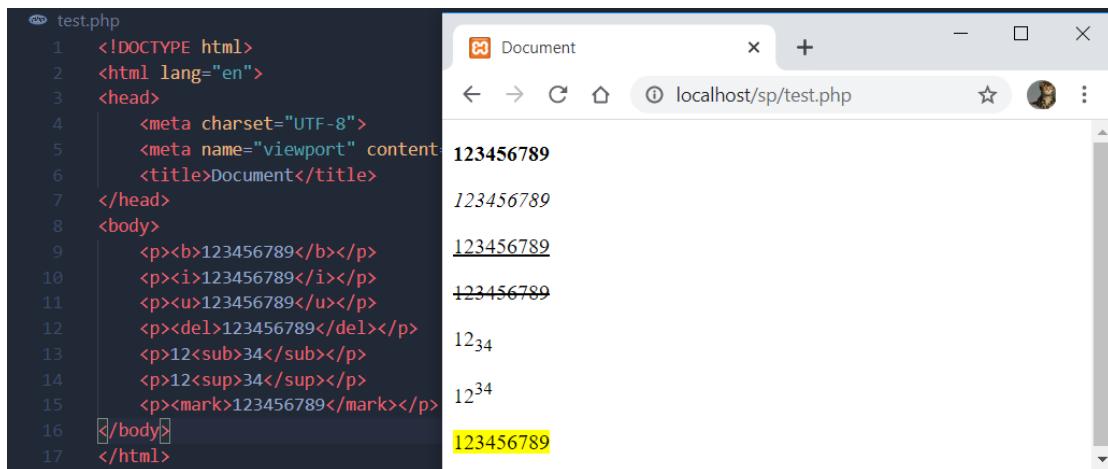
```

รูปที่ 2.19 แนวทางการใช้แท็กสำหรับการแสดงผลแบบแบ่งส่วนข้อความ

2.3.7 แท็กสำหรับจัดรูปแบบข้อความ

การจัดรูปแบบข้อความ เช่น ทำตัวหนา ตัวเอียง จีดเส้นใต้ เป็นต้น สำหรับในที่นี่จะกล่าวถึงเฉพาะแท็กที่น่าสนใจ ซึ่งเรามีโอกาสได้นำไปใช้งานจริงเท่านั้นคือ

	สำหรับทำ ตัวหนา (bold)	<sup>	สำหรับทำตัวยก (superscript)
<i>	สำหรับทำ ตัวเอียง (italic)	<sub>	สำหรับตัวห้อย (subscript)
<u>	สำหรับทำ จีดเส้นใต้ (underline)	<mark>	ทำไฮไลต์พื้นหลังข้อความ
	สำหรับจีดเส้นทับ เช่น Hello		



The screenshot shows a code editor on the left displaying an HTML file named 'test.php' with the following content:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <p><b>123456789</b></p>
10  <p><i>123456789</i></p>
11  <p><u>123456789</u></p>
12  <p><del>123456789</del></p>
13  <p>12<sub>34</sub></p>
14  <p>12<sup>34</sup></p>
15  <p><mark>123456789</mark></p>
16 </body>
17 </html>

```

To the right, a browser window titled 'Document' shows the rendered output:

123456789
123456789
123456789
~~123456789~~
12₃₄
12³⁴
123456789

รูปที่ 2.20 การใช้งานแท็กสำหรับจัดรูปแบบข้อความ

2.3.8 อักขระพิเศษของ HTML

อักขระบางตัว ถูกนำมาใช้สำหรับการกำหนดแท็ก HTML เช่น เครื่องหมาย < หรือ > และนอกจากนี้อักขระบางส่วนก็ไม่มีอยู่ในคีย์บอร์ด จึงไม่สามารถเขียนอักขระลงไปโดยตรงได้ แต่ต้องแทนด้วยสัญลักษณ์พิเศษ ซึ่งอันที่น่าสนใจคือ

ผลลัพธ์	อธิบาย	Entity Name
<	ช่องว่างไม่มีตัวคำ	&nbsp
<	น้อยกว่า	<
>	มากกว่า	>
&	แอนเพอร์แซนด์ (ampersand)	&
"	อัญประกาศ (double quotation mark)	"
'	อัญประกาศเดี่ยว (apostrophe)	'
¢	เงินเซ็น	¢
£	เงินปอนด์	£
¥	เงินเยน	¥
€	เงินยูโร	€
©	copyright	©
®	registered trademark	®

รูปที่ 2.21 อักขระพิเศษของ HTML

2.3.9 แท็กสำหรับแสดงคำอธิบาย

กรณีที่เราต้องการเขียนคำอธิบาย (Comment) ลงในโค้ด ให้กำหนดด้วยแท็ก `<!--...-->` โดยเว็บเบราว์เซอร์จะไม่นำคำอธิบายนี้ไปประมวลผล และไม่แสดงออกมากที่หน้าเว็บเพจให้เห็น หรืออาจเขียนครอบคำสั่ง HTML เพื่อไม่ให้เว็บเบราว์เซอร์ประมวลคำสั่งนั้นก็ได้ เช่น

```
<body>
    <!-- เริ่มต้นข้อความ -->
    123456789, <br>
    1011121314,
    ...
    <!-- สิ้นสุดข้อความ -->
</body>
```

2.3.10 การเชื่อมโยงแบบต่างๆ

การเชื่อมโยง (Link) ในเอกสาร HTML มีหลายลักษณะ เช่น การเชื่อมโยงระหว่างเพจทั้งที่อยู่ในเว็บไซต์เดียวกันและต่างเว็บไซต์กัน หรือการเชื่อมโยงสำหรับดาวน์โหลดไฟล์ เป็นต้น โดยจะกำหนดด้วยแท็ก `<a>` ในลักษณะดังนี้

`<a> แหล่งที่มาที่แสดงเป็นลิงก์ `

โดยแยกทริบิวต์ที่สำคัญของแท็ก `<a>` คือ

href	กำหนดเป้าหมายของการเชื่อมโยง เช่น เพจปลายทาง หรือไฟล์ เป็นต้น เช่น <code>DeveloperThai</code>
target	กำหนดวินโดว์เป้าหมายที่เปิดลิงก์นั้น โดยค่าที่กำหนดได้คือ <ul style="list-style-type: none"> • <code>_blank</code> เปิดวินโดว์หรือแท็บใหม่ • <code>_self</code> เปิดวินโดว์ปัจจุบันที่แสดงคณูน้อย • <code>_top</code> นำใช้ในกรณีที่แสดงเพจในเฟรม แล้วต้องการเปิดเพจนั้นในวินโดว์ปัจจุบันแบบเต็มเพจ หรือยกเลิกเฟรมนั้นเอง
id	กำหนดชื่ออ้างอิงกรณีที่จะเชื่อมโยงภายในเพจเดียวกัน

2.3.11 แท็กสำหรับแสดงรายการ

ลิสต์รายการ (List) ก็คือการแบ่งข้อมูลเป็นข้อๆอย่างๆ โดยมีการแท็กสำหรับกำหนดโครงร่างคือ

Unordered List	เป็นรูปแบบที่ไม่ใช้สัญลักษณ์การจัดเรียงรายการที่เป็นลำดับต่อเนื่อง เช่น <ul style="list-style-type: none"> • HTML
----------------	--

	<ul style="list-style-type: none"> ● CSS
Ordered List	<p>เป็นรูปแบบที่ใช้สัญลักษณ์การจัดเรียงรายการที่เป็นลำดับต่อเนื่อง เช่น</p> <ol style="list-style-type: none"> 1. HTML 2. CSS

รายการแต่ละแบบนั้นมีรายละเอียดปลีกย่อยที่แตกต่างกัน ดังต่อไปนี้

Unordered List

เนื่องจากรายการแบบนี้จะไม่ใช้สัญลักษณ์ที่เป็นลำดับแบบต่อเนื่อง จึงมักใช้กับรายการที่แยกเป็นข้อๆ โดยไม่สนใจลำดับก่อนหลัง โดยแท็กที่ใช้กำหนดโครงสร้างของ Unordered List คือ

	ใช้ในการกำหนดขอบเขตของลิสต์
	ใช้ในการกำหนดรายการต่างๆ ของลิสต์

```
<ul>
    <li>รูปภาพที่ 1</li>
    <li>รูปภาพที่ 2</li>
    ...
</ul>
```

เราสามารถกำหนดรูปแบบของสัญลักษณ์หน้ารายการได้ โดยใช้屬性 `type` ซึ่งมีรูปแบบให้เลือกดังนี้

<code>type="disc"</code>	จะเป็นวงกลมทึบ
<code>type="circle"</code>	เป็นวงกลมโปร่งใส
<code>type="square"</code>	เป็นรูปสี่เหลี่ยมทึบ

- ถ้าเรากำหนด屬性 `type="disc"` ให้แท็ก `` จะมีผลกับทุกรายการ
- ถ้าเรากำหนด屬性 `type="circle"` ให้แท็ก `` จะมีผลเฉพาะกับรายการนั้น
- หรือจะกำหนดทั้งที่แท็ก `` และ `` ได้ โดยรายการใดที่ต้องการรูปแบบเฉพาะให้ระบุที่แท็ก `` แต่ถ้าไม่ระบุจะเป็นรูปแบบตามที่กำหนดที่แท็ก ``

Ordered List

รายการแบบ Ordered List จะใช้สัญลักษณ์ที่มีลักษณะการเรียงลำดับที่แน่นอน เช่น 1 – 5 หรือ a – f เป็นต้น โดยแท็กที่เกี่ยวข้องกับลิสต์แบบนี้คือ

	ใช้ในการกำหนดขอบเขตของลิสต์ มี屬性 <code>type</code> ที่น่าสนใจคือ
------	--

	● Start สำหรับกำหนดค่าเริ่มต้นของลำดับรายการ
	ใช้ในการกำหนดรายการของลิสต์

เราสามารถเลือกใช้สัญลักษณ์สำหรับ Ordered List ได้หลายอย่าง เช่น แบบตัวเลข ตัวอักษรภาษาอังกฤษ เป็นต้น โดยกำหนดผ่านแอ็ตทริบิวต์ type เช่นเดียวกับ Unordered List ซึ่งสามารถใช้สัญลักษณ์หน้ารายการเป็นรูปแบบต่างๆ ดังนี้

type="1"	เป็นแบบตัวเลข เช่น 1, 2, 3
type="a"	เป็นตัวอักษรภาษาอังกฤษแบบพิมพ์เล็ก เช่น a, b, c
type="A"	เป็นตัวอักษรภาษาอังกฤษแบบพิมพ์ใหญ่ เช่น A, B, C
type="i"	ตัวเลขโรมันแบบพิมพ์เล็ก เช่น i, ii, iii
type="I"	ตัวเลขโรมันแบบพิมพ์ใหญ่ เช่น I, II, III

2.4 CSS Basic & Selector

CSS (Cascading Style Sheets) เป็นคำสั่งที่ใช้สำหรับจัดรูปแบบเอกสารเว็บเพจในลักษณะต่างๆ เช่น สี พื้นหลัง เส้นขอบ โครงร่าง หรือตัวอักษร เป็นต้น แม้ว่าใน HTML นั้นจะมีทั้งแท็กและแอ็ตทริบิวต์บางส่วนที่สามารถใช้จัดรูปแบบเอกสารอยู่แล้ว แต่ก็ยังไม่ครอบคลุมการใช้งานครบถ้วน นอกเหนือจากนี้ก็ยังขาดความยืดหยุ่นและไม่สามารถจัดรูปแบบที่ซับซ้อนได้ ซึ่งในปัจจุบันเราจะเน้นการจัดรูปแบบด้วย CSS มากกว่าที่จะใช้แท็กและแอ็ตทริบิวต์ของ HTML ดังนั้น จึงเกิดการแยกหน้าที่กันอย่างชัดเจนคือ HTML ใช้สำหรับการจัดโครงร่างของเว็บเพจ ส่วน CSS ใช้สำหรับจัดรูปแบบของเว็บเพจ สำหรับในโครงงานนี้เราจะได้ศึกษาการจัดรูปแบบเอกสารด้วย CSS ในเบื้องต้น

2.4.1 พร็อปเพอร์ตี้ของ CSS

พร็อปเพอร์ตี้ (Property) ของ CSS คือสิ่งที่ใช้กำหนดคุณลักษณะต่างๆ ของอิเลมเม้นต์ HTML เช่น ทำให้ข้อความสีฟ้า + พื้นหลังเป็นสีเหลือง + ใช้ฟอนต์ชนิด Tahoma หรือทำให้มีขนาดที่ต้องการ เป็นต้น สำหรับหลักการเบื้องต้นเกี่ยวกับพร็อปเพอร์ตี้ที่เราควรทราบ จะแยกพิจารณาเป็นประเด็นต่างๆ ดังต่อไปนี้

2.4.1.1 การกำหนดค่าของพร็อปเพอร์ตี้

รูปแบบการกำหนดค่าของพร็อปเพอร์ตี้จะมี 2 ส่วนคือ ชื่อพร็อปเพอร์ตี้ และค่าของมัน โดยคั่นระหว่างชื่อและค่าด้วยเครื่องหมาย Colon (:) คือ

ชื่อพร็อปเพอร์ตี้: ค่าที่กำหนด

กรณีที่มีมากกว่า 1 พร็อปเพอร์ตี้ ให้คั่นแต่ละพร็อปเพอร์ตี้ด้วยเครื่องหมาย Semicolon (;) ในรูปแบบดังต่อไปนี้

ชื่อพร็อปเพอร์ตี้ 1	: ค่าที่กำหนด ; ชื่อพร็อปเพอร์ตี้ 2 : ค่าที่กำหนด ; ...
---------------------	---

ซึ่งแนวทางการกำหนดพร็อปเพอร์ตี้จะเป็นดังนี้

color : white ; background-color : black ; front-size : 12pt ;

หรือถ้าเราไม่เขียนรวมไว้ในบรรทัดเดียวกัน เราอาจเขียนแยกกันไว้คันลับบรรทัด กันก็ได้ เนื่องจากมีเครื่องหมาย ; คั่นระหว่างพร็อปเพอร์ตี้อยู่แล้ว เช่น

color : white ;

background-color : black ;

front-size : 12pt ;

2.4.1.2 พร็อปเพอร์ตี้พื้นฐานที่ควรรู้จัก

ใน CSS นั้นมีพร็อปเพอร์ตี้ที่ใช้กำหนดครุปแบบเป็นจำนวนมาก ซึ่งเราจะได้ศึกษา ต่อเนื่องกันไปเรื่อยๆ ในเบื้องต้น เราควรรู้จักกับพร็อปเพอร์ตี้พื้นฐานบางส่วนของ CSS ล่วงหน้าไว้ ก่อน ซึ่งมีดังนี้

color	กำหนดสีข้อความ โดยในเบื้องต้นให้ระบุค่าเป็นชื่อสีในภาษาอังกฤษไป ก่อน เช่น green, blue, red, gray เป็นต้น
background-color	กำหนดสีพื้นหลัง โดยระบุค่าเป็นชื่อสีเช่นเดียวกับ color
front-size	กำหนดขนาดของฟอนต์ ซึ่งในเบื้องต้นให้กำหนดค่าเป็นตัวเลข ตามด้วย หน่วย pt หรือ px โดยตัวเลขและหน่วยให้เขียนติดกัน
front-weight	กำหนดขนาดหนาแน่นักของฟอนต์ มักใช้เมื่อต้องการทำให้ข้อความเป็น ตัวหนา โดยระบุค่าเป็น bold
front-style	กำหนดรูปแบบฟอนต์ โดยทั่วไปมักกำหนดเมื่อต้องการทำให้ข้อความ เป็นตัวเอียง โดยระบุค่าเป็น italic
Text-decoration	สำหรับตกแต่งข้อความ มักใช้เมื่อต้องการขีดเส้นใต้ โดยระบุค่าเป็น underline

2.4.2 ขอบเขตของการกำหนดสไตล์แบบต่างๆ

การกำหนดสไตล์แบ่งออกเป็น 3 แบบ คือ

- สไตล์แบบ Inline จะมีผลเฉพาะกับอิเลเมนต์ที่กำหนดสไตล์กับมัน
- สไตล์แบบ Internal จะมีผลกับทุกอิเลเมนต์ภายในพจน์นั้น ที่ตรงกับข้อกำหนด

- ตัวลักษณ์แบบ External นำเข้าสู่ตัวจากไฟล์ภายนอก และจะมีผลกับทุกอิเลี่ยมต์ภายในเพจนั้น ที่ตรงกับข้อกำหนด

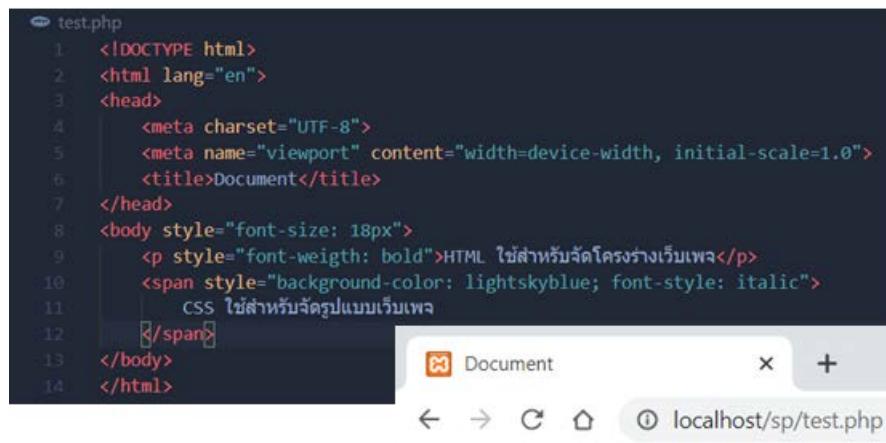
สำหรับรายละเอียดการกำหนดตัวลักษณ์แต่ละแบบมีดังต่อไปนี้

2.4.2.1 การกำหนดตัวลักษณ์แบบ Inline

ตัวลักษณ์แบบ Inline คือการกำหนดรูปแบบเอาไว้ที่แท็กที่เราต้องการจัดรูปแบบ โดยระบุพร้อมเพอร์ตี้และค่าของแต่ละทริบิวต์ style ซึ่งรูปแบบคือ

```
<แท็ก style="พร็อปเพอร์ตี้-1 ; พร็อปเพอร์ตี้-2 ; ค่าที่กำหนด ; ...">
```

แนวทางการกำหนดตัวลักษณ์แบบ inline เช่น



```

test.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body style="font-size: 18px">
9      <p style="font-weight: bold">HTML ใช้ส่วนรับจัดโครงสร้างเว็บ</p>
10     <span style="background-color: lightskyblue; font-style: italic">
11         CSS ใช้ส่วนรับจัดรูปแบบเว็บ
12     </span>
13 </body>
14 </html>

```

The screenshot shows a code editor with the file 'test.php' containing the above code. To the right, a browser window titled 'Document' is open, showing the rendered HTML. The text 'HTML ใช้ส่วนรับจัดโครงสร้างเว็บ' is in bold, and the word 'CSS ใช้ส่วนรับจัดรูปแบบเว็บ' is in italics and has a light blue background.

HTML ใช้ส่วนรับจัดโครงสร้างเว็บ

CSS ใช้ส่วนรับจัดรูปแบบเว็บ

รูปที่ 2.22 แนวทางการกำหนดตัวลักษณ์แบบ Inline

2.4.2.2 การกำหนดตัวลักษณ์แบบ Internal

ตัวลักษณ์แบบ Internal จะกำหนดเอาไว้ที่ส่วนหัวของเอกสาร เพื่อให้มีผลกับหลายๆ อิเลี่ยมต์ โดยไม่ต้องเปลี่ยนช้ากันอีก ดังต่อไปนี้

```

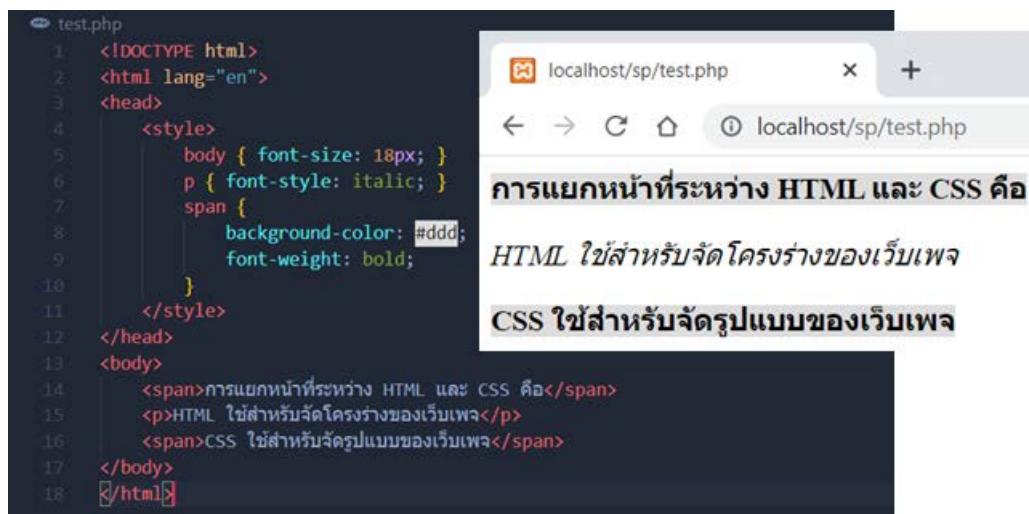
<style>
    ชิ้นเล็กเดอร์ {
        พร็อปเพอร์ตี้-1 : ค่าที่กำหนด ;
        พร็อปเพอร์ตี้-2 : ค่าที่กำหนด ;
        ...
    }

```

```
</style>
```

สิ่งที่ควรรู้เพิ่มเติมเกี่ยวกับสไต์ล์แบบ Internal คือ

- สไต์ล์แบบ Internal ต้องกำหนดด้วยแท็ก `<style>` และให้อยู่ระหว่างแท็ก `<head>...</head>`
- ซีเลกเตอร์ (Selector) เป็นตัวใช้กำหนดว่า จะให้สไต์ล์นั้นมีผลกับอิลิเม้นต์ใด โดยมีซีเลกเตอร์หลายรูปแบบ เช่น Type, ID, Class เป็นต้น ซึ่งรายละเอียดเกี่ยวกับซีเลกเตอร์แต่ละแบบ จะกล่าวถึงในหัวข้อต่อๆ ไป
- พร็อปเพอร์ตี้และค่าของมัน ต้องกำหนดไว้ในวงเล็บ {} เช่น
- เราอาจเขียนพร็อปเพอร์ตี้รวมไว้ในบรรทัดเดียวได้ดังโค้ดที่แล้ว หรือแยกบรรทัดกันก็ได้

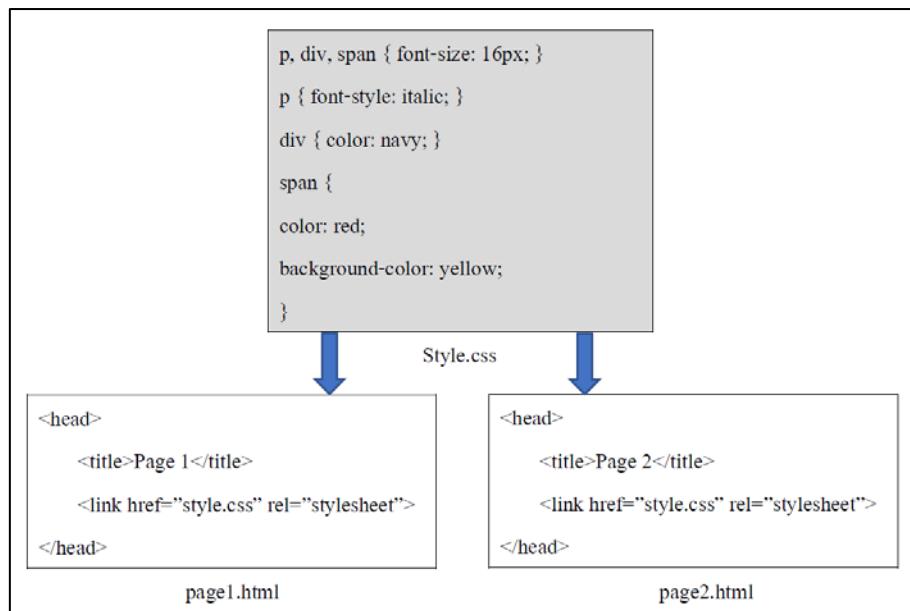


รูปที่ 2.23 แนวทางการกำหนดสไต์ล์แบบ Internal

2.4.2.3 การกำหนดสไต์ล์แบบ External

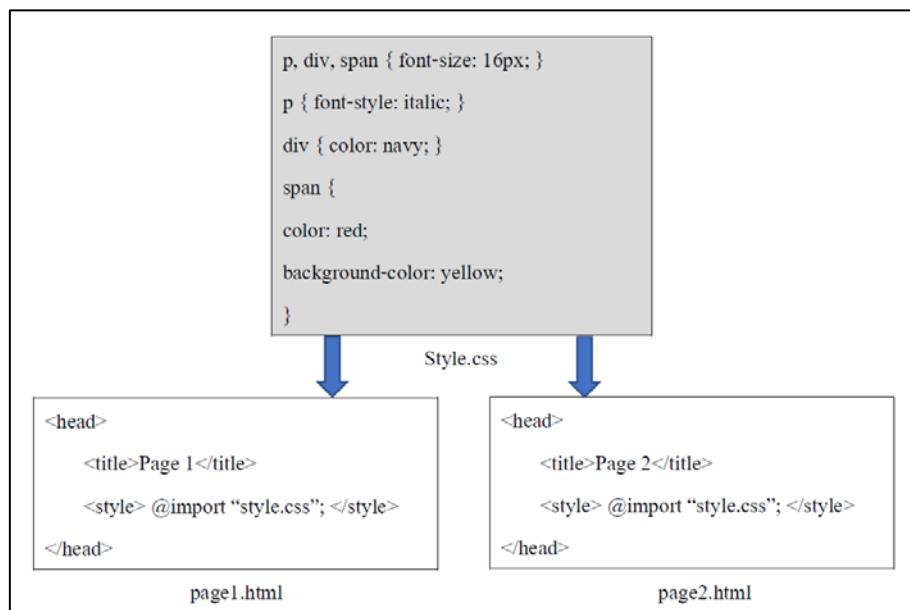
สไต์ล์แบบ External คือการแยกส่วนสไต์ล์ออกไปไว้ที่ไฟล์ภายนอกต่างหาก ซึ่งมักใช้ในกรณีที่เราจำเป็นต้องใช้สไต์ล์แบบเดียวกันในหลายๆ เพจ โดยแยกเฉพาะส่วนสไต์ล์ที่ซ้ำกันไปไว้ที่ไฟล์ภายนอก แล้วเมื่อต้องการใช้ที่เพจใด ก็ทำการเชื่อมโยงกับไฟล์นั้น ซึ่งมีให้เลือก 2 วิธีคือ

- วิธีที่ 1 เชื่อมโยงโดยว่างแท็ก `<link>` ไว้ที่เพจที่ต้องการใช้ส์ไฟล์ ดังภาพ



รูปที่ 2.24 การเชื่อมโยงด้วยแท็ก `<link>`

- วิธีที่ 2 เชื่อมโยงด้วยคีย์เวิร์ด `@import` โดยว่างไว้ที่แท็ก `<style>` ของเพจที่ต้องการใช้ส์ไฟล์ ดังภาพ



รูปที่ 2.25 การเชื่อมโยงคีย์เวิร์ด `@import`

สิ่งที่ควรรู้เพิ่มเติมเกี่ยวกับสีไอล์แบบ External กือ

- โภคดของส่วนสีไอล์ที่แยกไปสร้างไว้ที่ไฟล์ภายนอกนั้น ต้องไม่มีแท็ก `<style>` รวมอยู่ด้วย
- ไฟล์ของสีไอล์ภายนอกนั้น ให้กำหนดส่วนขยายเป็น `.css`
- สำหรับกรณีการใช้แท็ก `<link>` มีแอ็ตทริบิวต์ที่สำคัญ 2 อย่างคือ
 - `href` สำหรับการกำหนดชื่อไฟล์ หากไฟล์ CSS กับไฟล์เว็บเพจที่เรียกใช้งานอยู่คุณจะได้รีกอรี ให้ระบุเส้นทางลงไปด้วย เช่น `href="../src/css/style.css"` ซึ่งกรณีใช้ `@import` ก็ทำเช่นเดียวกัน
 - `rel` สำหรับกำหนดชนิดข้อมูลที่เราจะเข้ามายังไฟล์นี้ โดยในกรณีไฟล์ CSS ให้กำหนดค่าเป็น `rel="stylesheet"`
- เราสามารถนำไฟล์ CSS จากภายนอกเข้ามาในเพจเดียวกันจำนวนกี่ไฟล์ก็ได้ โดยแต่ละไฟล์ให้กำหนดด้วยแท็ก `<link>` หรือคีย์เวิร์ด `@import` แยกจากกัน เช่น

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

2.4.3 รูปแบบซีเลกเตอร์ที่น่าสนใจ

กรณีที่เราใช้สีไอล์แบบ Internal หรือ External สิ่งที่ต้องกำหนดเอาไว้หน้าบล็อกของรายการพร้อมเพอร์ตี้จะเรียกว่า ซีเลกเตอร์ (Selector) ซึ่งรูปแบบคือ

```
ซีเลกเตอร์ {
    พร้อมเพอร์ตี้-1 : ค่าที่กำหนด ;
    พร้อมเพอร์ตี้-2 : ค่าที่กำหนด ;
    ...
}
```

โดยซีเลกเตอร์จะเป็นตัวกำหนดลักษณะของอิลิเม้นต์ที่จะถูกเลือกมาจัดรูปแบบดังกล่าว ซึ่งมีรูปแบบของซีเลกเตอร์ให้เราเลือกใช้มากmany แค่ไหนก็ได้ ถ้าต้องการอ่านเพิ่มเติม ก็สามารถหาได้ในที่นี่ [รายละเอียดเพิ่มเติม](#)

ชีลีกเตอร์แบบ Universal

Universal Selector จะเขียนแทนด้วยเครื่องหมาย * เป็นชีลีกเตอร์พื้นฐานที่สุดของ CSS ใช้สำหรับกำหนดให้อลิเมนต์ทั้งหมดในเอกสารมีรูปแบบตรงตามที่ระบุ เช่น หากต้องการให้ข้อความทุกอลิเมนต์เป็นสีเขียว ก็กำหนดดังนี้

```
<style>
  * { color : green ; }
</style>
```

ซึ่งในทางปฏิบัติ เราไม่ได้ใช้ชีลีกเตอร์อันนี้กันบ่อยนัก ยกเว้นจะนำไปประยุกต์ใช้ร่วมกับชีลีกเตอร์อื่นๆ ในแบบ Multiple Selector ดังที่จะกล่าวถึงในหัวข้อต่อๆ ไป

ชีลีกเตอร์แบบ Tag

Tag Selector เป็นการนำชื่อแท็กมากำหนดเป็นชีลีกเตอร์ทำให้สไตล์ที่เรากำหนด มีผลกับอลิเมนต์ของแท็กชนิดนั้นๆ ทุกอันที่อยู่ในไฟล์ เช่น

```
test.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <style>
5      p { color: blue; }
6      div {
7        color: white;
8        background-color: black;
9      }
10     span {
11       font-size: 18px;
12       font-weight: bold;
13     }
14   </style>
15 </head>
16 <body>
17   <p>ข้อความนี้อยู่ใน p</p>
18   <div>ข้อความนี้อยู่ใน div</div> <br>
19   <span>ข้อความนี้อยู่ใน span</span> <br><br>
20   <div>ข้อความนี้อยู่ใน div</div>
21   <p>ข้อความนี้อยู่ใน p</p>
22 </body>
23 </html>
```

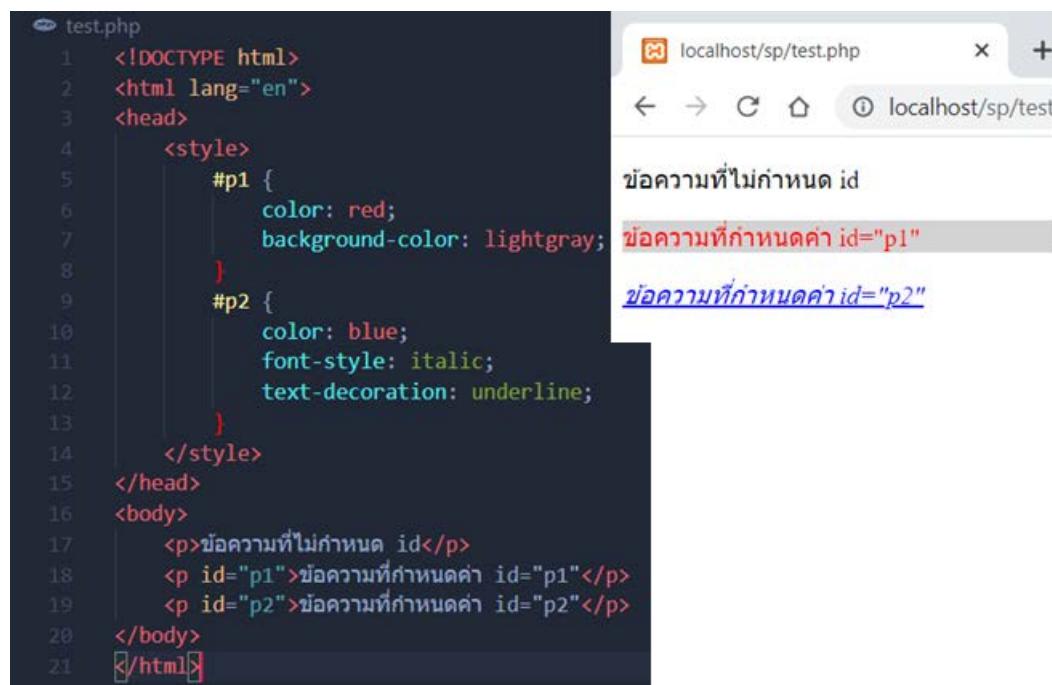
รูปที่ 2.26 ชีลีกเตอร์แบบ Tag

ชีลีกเตอร์แบบ ID

แอตทริบิวต์ ID ใช้สำหรับกำหนดชื่ออ้างอิงให้กับแท็กต่างๆ เช่น <div id="title"> หรือ <p id="warning"> เป็นต้น แต่ภายในเพจเดียวกัน ค่า id ของแต่ละแท็กต้องไม่ซ้ำกัน และจากการที่ค่า id เป็นชื่ออ้างอิงของแต่ละแท็ก จึงสามารถนำมาระบุกำหนดเป็นชีลีกเตอร์อีกชนิดหนึ่ง ดังต่อไปนี้

```
#id {
    พร้อมเพร็ช: ค่าที่กำหนด ;
    ...
}
```

เราต้องเขียนเครื่องหมาย # กำกับที่หน้าชื่อ id เสมอ เพื่อบ่งชี้ว่าเป็น id Selector โดยสไตล์ที่เรากำหนดจะมีผลเฉพาะกับแท็กที่มีค่า id ตรงกับชีลีกเตอร์ แต่ค่า id ที่กำหนดให้แก่แอตทริบิวต์ของแท็ก HTML ไม่ต้องมีเครื่องหมาย # นำหน้า เช่น



รูปที่ 2.27 ชีลีกเตอร์แบบ ID

ถ้าเรานำหลายๆ คำารวณกันเป็นค่า id ก็สามารถใช้เครื่องหมาย – คั่นระหว่างคำได้ เช่น

```
<style>
    #first-div { font-weight: bold; }
    #last-div { text-decoration: underline; }
```

```
</style>

<body>

    <div id="first-div">Hello</div>

    <div id="last-div">Good bye</div>

</body>
```

ชีลีกเตอร์แบบ Class

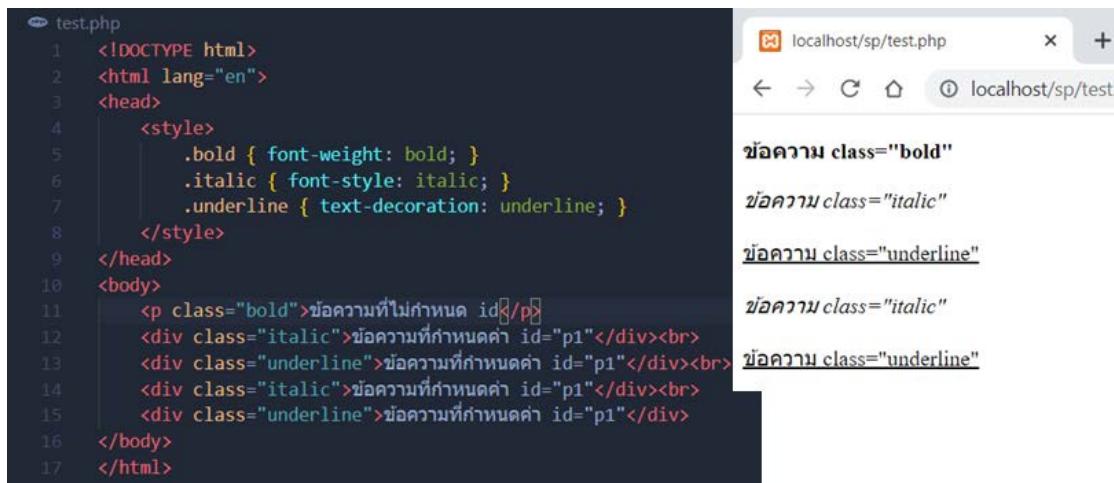
แท็กส่วนใหญ่ของ HTML สามารถกำหนดแอ็ตทริบิวต์ชื่อ class ให้กับมัน ได้ เช่น

```
<div class="title"> หรือ <p class="message">
```

ดังนั้น จึงสามารถนำแอ็ตทริบิวต์ class มาสร้างเป็นชีลีกเตอร์ได้อีกชนิดหนึ่ง ในลักษณะดังนี้

```
.ชื่อคลาส {
    พร้อมเพรียงตัวที่กำหนด ;
    ...
}
```

ข้อกำหนดสำคัญของ Class Selector คือ จะต้องมีจุด(.) ไว้หน้าชื่อคลาส แต่ชื่อคลาสที่กำหนดให้แก่แอ็ตทริบิวต์ของ HTML ไม่ต้องมีจุดนำหน้า เช่น



```
test.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <style>
5          .bold { font-weight: bold; }
6          .italic { font-style: italic; }
7          .underline { text-decoration: underline; }
8      </style>
9  </head>
10 <body>
11     <p class="bold">ข้อความที่ไม่กำหนด id</p>
12     <div class="italic">ข้อความที่กำหนดค่า id="p1"</div><br>
13     <div class="underline">ข้อความที่กำหนดค่า id="p1"</div><br>
14     <div class="italic">ข้อความที่กำหนดค่า id="p1"</div><br>
15     <div class="underline">ข้อความที่กำหนดค่า id="p1"</div>
16 </body>
17 </html>
```

รูปที่ 2.28 ชีลีกเตอร์แบบ Class

เราสามารถกำหนดมากกว่า 1 คลาสให้กับแท็กเดียวกัน โดยระบุเป็นค่าของแอ็ตทริบิวต์ class เดียวกันแล้วใช้ช่องว่างคั่นระหว่างชื่อคลาส เช่น <div class="class1 class2 class3"> ข้อดีของการใช้หลายคลาสก็คือ กันณีที่เราแบ่งส่วนลักษณะเป็นรูปแบบย่อยๆ และถ้าต้องการให้อิเลิมอนต์นั้นตรงกับรูปแบบของคลาสใดบ้าง ก็เลือกคลาสเหล่านั้นมาใช้ร่วมกันได้เลย

ชีลีกเตอร์แบบ Tag.Class

ชีลีกเตอร์แบบ Tag.Class เป็นการนำเอาชีลีกเตอร์ 2 แบบมาใช้ร่วมกันคือ Tag Selector และ class Selector โดยมีรูปแบบดังนี้

```
แท็ก.ชื่อคลาส {  
    พร้อมเพอร์ติ้ง: ค่าที่กำหนด;  
    ...  
}
```

ชีลีกเตอร์แบบ Tag.Class นี้จะมีผลเฉพาะกับแท็กที่กำหนดชื่อคลาสตรงกับที่ระบุเท่านั้น

ชีลีกเตอร์แบบ Union

Union Selector เป็นการนำเอาชีลีกเตอร์หลายๆ รูปแบบมาใช้ส่วนลักษณะร่วมกันคือ

```
ชีลีกเตอร์1, ชีลีกเตอร์2, ... {  
    พร้อมเพอร์ติ้ง: ค่าที่กำหนด;  
    ...  
}
```

ชีลีกเตอร์ที่จะนำมาใช้ อาจเป็นชนิดเดียวกันหรือต่างชนิดทั้ง Tag, ID, Class, Selector หรืออื่นๆ โดยคั่นแต่ละชีลีกเตอร์ด้วยเครื่องหมาย , ซึ่ง Union Selector จะช่วยให้เรากำหนดส่วนร่วมเดียวกันสำหรับอิเลิมอนต์หลายตัว โดยไม่ต้องเขียนโค้ด CSS ซ้ำซ้อนกัน

2.5 JavaScript Basic

ในปัจจุบัน จาสวัสดิ์คือเป็นองค์ประกอบที่สำคัญอย่างหนึ่งในการสร้างเว็บที่เรารู้สึกขาดไม่ได้กับ HTML และ CSS เนื่องจากลิสต์ที่กำหนดด้วย HTML และ CSS นั้น จะเป็นรูปแบบที่ติดต่อไม่สามารถเปลี่ยนแปลงตามเงื่อนไขได้ การสร้างเว็บจึงทำได้ในขอบเขตที่จำกัด แต่หากนำจาสวัสดิ์มาใช้สามารถกำหนดวิธีการทำงานได้ตามเงื่อนไขที่เราต้องการ และแม้ว่าจาสวัสดิ์จะมีองค์ประกอบค่อนข้างมาก แต่ในโครงงานนี้จะเน้นเฉพาะลักษณะที่เราจำเป็นต้องทราบในเบื้องต้น และสามารถนำไปใช้ร่วมกับการพัฒนาเว็บเพจทั่วๆ ไปได้สำหรับในหัวข้อนี้จะกล่าวถึงพื้นฐาน

ที่เราควรรู้จัก เช่น การเขียนภาษาสคริปต์ในเอกสาร HTML การแสดงผล การกำหนดตัวแปรและข้อมูล รวมถึงการใช้โอลีโอเรเตอร์ในรูปแบบต่างๆ

2.5.1 คำส่วนของภาษาสคริปต์

ในภาษาสคริปต์มีคำส่วน (Reserved Words หรือ Keywords) เช่นเดียวกับภาษาอื่นๆ สำหรับใช้เป็นคำสั่งหลักๆ เพื่อกำหนดหรือควบคุมการทำงาน ซึ่งเราไม่สามารถนำคำเหล่านี้ไปตั้งเป็นชื่อตัวแปร พังก์ชัน คลาส หรืออื่นๆ ได้ โดยคำส่วนทั้งหมดในภาษาสคริปต์ มีดังนี้

abstract	delete	function	null	throw
boolean	do	goto	package	throws
break	double	if	private	transient
byte	else	implements	protected	true
case	enum	import	public	try
catch	export	in	return	typeof
char	extends	instanceof	short	var
class	false	int	static	void
const	final	interface	super	volatile
continue	finally	long	switch	while
debugger	float	native	synchronized	with
default	for	new	this	

รูปที่ 2.29 คำส่วนของภาษาสคริปต์

สำหรับคำส่วนเหล่านี้ ส่วนใหญ่เป็นข้อกำหนดดั้งเดิมของภาษาสคริปต์ แต่บางคำก็เพิ่งจะเริ่มใช้ในເອັກນາສคริปต์ 5 และ 6 ส่วนคำที่ยังไม่ใช้อย่างเป็นทางการ เช่น await, enum ผู้เขียนไม่ได้นำมารวมไว้ด้วย

2.5.2 องค์ประกอบเบื้องต้นของการเขียนໂຄດ

การเขียนໂຄດภาษาสคริปต์นี้ มีองค์ประกอบพื้นฐานบางอย่างที่เราต้องใช้งานกันอยู่ตลอด เราจึงควรรู้จักกับสิ่งเหล่านี้น้อาไว้ล่วงหน้า ดังรายละเอียดต่อไปนี้

2.5.2.1 การเขียนໂຄດภาษาสคริปต์ในเอกสาร HTML

การเขียนภาษาสคริปต์จะกำหนดไว้ระหว่างแท็ก `<script>...</script>` สำหรับตำแหน่งในการเขียนนั้นเขียนกับการเรียกใช้งานสคริปต์เป็นหลัก ซึ่งสามารถจำแนกหลักการทั่วไปของการวางแท็ก `<script>` ในเอกสาร HTML ได้ดังนี้

- ถ้าเป็นสิ่งที่ต้องการใช้ร่วมกันหลายๆ ที่ภายในไฟล์ เช่น ตัวแปรแบบ global หรือฟังก์ชัน เรานิยมกำหนดสคริปต์เหล่านี้ไว้ที่ส่วนหัวของไฟล์ระหว่างแท็ก <head>...</head> ของ HTML เช่น

```
<!DOCTYPE html>
<html>
<head>
<script>
    var x = 100;
    function calculate () {
        ...
    }
</script>
</head>
</html>
```

- ถ้าสคริปต์เป็นคำสั่งที่ต้องการแสดงหรือประมวลผล ณ จุดใดจุดหนึ่งภายในเอกสาร ก็ให้วางแท็กนี้ไว้ภายใน <body>...</body> ณ จุดที่ต้องการ

```
<body>
<script>
    document.write("Hello World");
</script>
</body>
```

- ภายในเอกสารเดียวที่นั้น จะมีจำนวนแท็ก <script>...</script> กี่อันก็ได้ และสามารถเขียนสลับกับแท็กของ HTML โดยโค้ดที่อยู่ในแท็ก <script> คนละ อัน ก็เชื่อมโยงถึงกันได้ เช่น

```
<head>
<script>
    var x = 100 ;
    const pi = 3.141
</script>
</head>
<body>
```

```

<script>
    var y = 200 ;
    var z = x + y + pi ;
</script>

<h1> z =
<script>
    document.write(z) ;
</script>
</h1>

</body>

```

อย่างไรก็ตาม เราไม่จำเป็นต้องกำหนดค่าสิ่งจำพวกคริปต์ด้วยแท็ก `<script>` เสมอไป แต่อาจเขียนให้เป็นค่าแอ็ตทริบิวต์ของแท็ก HTML ก็ได้ เช่นกรณีต่อไปนี้

```

<a href="javascript : alert('Hello')>Hello</a>
<button type="button" onclick="alert('Hello')>Hello</button>

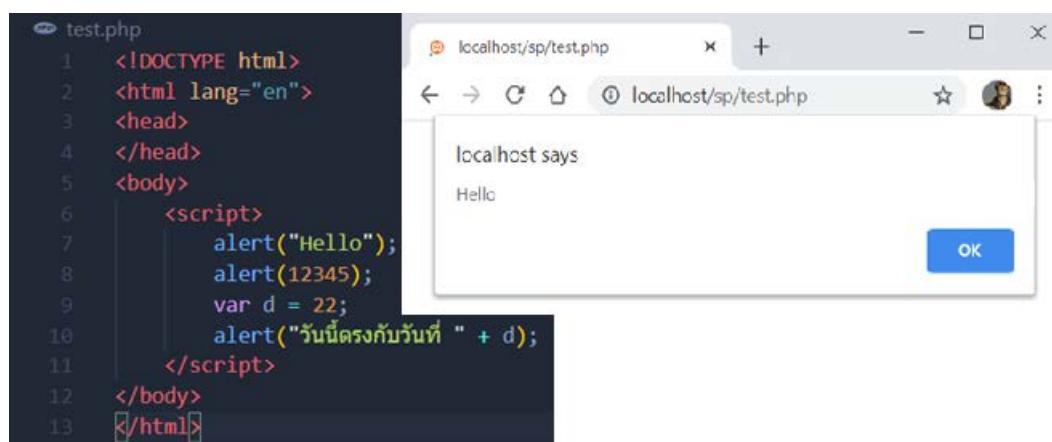
```

นอกจากนี้ยังมีวิธีการอื่นๆ อีกหลายแบบ

2.5.2.2 การแสดงข้อความแจ้งเตือนด้วย `alert()`

ฟังก์ชัน `alert()` ใช้ในการแจ้งเตือน หรือการแสดงข้อมูลบางอย่างแก่ผู้ใช้งาน แต่อาจมีลักษณะที่แตกต่างกันในแต่ละเบราว์เซอร์ ดังภาพถัดไปซึ่งเป็น `alert` ของ Firefox

สำหรับวิธีการใช้งาน เราเพียงแต่ระบุข้อมูลที่ต้องการแสดงให้แก่ฟังก์ชัน `alert()` ซึ่งถึงที่จะแสดงนั้นอาจเป็นข้อความ ตัวเลข ตัวแปร การคำนวณ หรืออื่นๆ ก็ได้ เช่น



รูปที่ 2.30 การแสดงข้อความแจ้งเตือนด้วย `alert()`

2.5.2.3 การแสดงข้อมูลบนหน้าเพจด้วย document.write()

คำสั่ง document.write() ใช้สำหรับการเขียนข้อมูลบางอย่างแทรกลงในเว็บเพจ ณ ตำแหน่งที่วางคำสั่งนี้ลงไป หรือจุดที่เรียกใช้คำสั่งนี้ โดยข้อมูลที่เขียนลงไปอาจเป็นข้อความ ตัวเลข ตัวแปร หรือแท็ก HTML และจะถูกป้อนส่วนหนึ่งของเอกสาร ซึ่งหากเป็นแท็ก HTML ก็จะปรากฏผลไปตามลักษณะของแท็กนั้น และถ้าต้องการบรรทัดใหม่ ก็ให้เขียนแท็ก
 ออกไป เช่น โค้ดต่อไปนี้

```

test.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  </head>
5  <body>
6      ข้อความนี้เขียนใน HTML โดยตรง
7      <script>
8          document.write("<br><br>");
9          document.write("ข้อความนี้เขียนด้วย Javascript");
10         document.write("<hr>");
11         document.write(
12             '<div style="font-weight:bold; color:red">' +
13             "การทำเงินประกอบด้วย:</div>');
14         document.write("<ul>");
15         document.write("<li>HTML</li>");
16         document.write("<li>CSS</li>");
17         document.write("<li>Javascript</li>");
18         document.write("</br>");
19     </script>
20 </body>
21 </html>

```

รูปที่ 2.31 การแสดงข้อมูลบนหน้าเพจด้วย document.write()

2.5.2.4 การเขียนคำอธิบายโค้ดของภาษาสคริปต์

คำอธิบายโค้ด (Comment) เป็นข้อความที่เขียนแทรกไว้ในโค้ดของโปรแกรม สำหรับอธิบายหลักการทำงานหรือแสดงข้อความใดๆ ก็ตามที่เราต้องการ ซึ่งอาจเขียนไว้เพื่อเดือนความทรงจำเมื่อเรากลับมาอ่านโค้ดในภายหลัง หรือเมื่อส่งต่อให้แก่ผู้อื่น ก็จะเข้าใจข้อกำหนดต่างๆ ของส่วนนั้นได้เป็นต้น โดยการเขียนคำอธิบายในภาษาภาษาสคริปต์มี 2 รูปแบบดังนี้

Line Comment

Line Comment (หรือ Single-Line Comment) นั่นใช้กับการเขียนคำอธิบายสั้นๆ ให้จบภายในหนึ่งบรรทัด โดยใช้เครื่องหมาย Slash จำนวน 2 อัน (//) วางไว้ ณ จุดที่ต้องการแทรก จากนั้นก็เขียนข้อความต่อท้ายเครื่องหมาย // ได้เลย ซึ่งตั้งแต่เครื่องหมาย // ไปจนถึงสุดบรรทัดนั้น

จะถือว่าเป็นคำอธิบายและไม่นำมาประมวลผลด้วย แม้สิ่งที่ไว้เขียนตามหลัง // จะเป็นคำสั่งจาก скриปต์ก็ตาม เช่น

```
//document.write("x"); บรรทัดนี้ไม่ถูกประมวลผล แม้จะเป็นคำสั่งจาก скриปต์  
document.write("x"); //แต่บรรทัดนี้ถูกประมวลผล เพราะเขียนไว้ก่อน Comment
```

Block Comment

Block Comment (หรือ Multi-Line Comment) มักใช้ในกรณีที่จะเปลี่ยนคำอธิบาย
ภาษา หลายบรรทัดซึ่งรูปแบบของวิธีนี้คือ

- กำหนดจุดเริ่มต้นของบล็อกด้วยเครื่องหมาย Slash ตามด้วย Asterisk (*)
- จากนั้นก็เขียนคำอธิบายลงไปตามต้องการ จะมีกับบรรทัดก็ได้
- เมื่อจบคำอธิบาย ให้กำหนดสิ้นสุดบล็อกด้วยเครื่องหมาย Asterisk (*)

```
/* 2 บรรทัดต่อไปนี้ไม่ถูกประมวลผล แม้จะเป็นคำสั่งจาก скриปต์  
document.write("Hello");  
document.write("World");  
*/  
document.write("สวัสดี ชาวโลก"); /*แต่บรรทัดนี้ถูกประมวลผล เพราะไม่มีอยู่ใน Comment */
```

นอกจากนี้ ยังสามารถเขียน Block Comment แทรกไว้ในช่วงระหว่างโค้ดก็ได้
โดยส่วนใหญ่จะเป็นลักษณะการทำให้โค้ดส่วนนั้นไม่ถูกประมวลผล เช่น

```
If (withdraw % 100 != 0 /* || withdraw > 20000 */ {  
    //..  
}
```

2.5.2.5 Statement และเครื่องหมายสิ้นสุดคำสั่ง

Statement คือคำสั่งที่สมบูรณ์ซึ่งโปรแกรมสามารถนำไปใช้ในการประมวลผลเพื่อ
กระทำการบางอย่าง ได้ โดยเราแบ่ง Statement ออกเป็น 2 ลักษณะคือ

- Single Statement คือ Statement ที่มีเพียงคำสั่งเดียว
- Compound Statement คือ Statement ที่มีหลายคำสั่ง หรือมี Statement ข้อๆ
หลายอันรวมกันนั่นเอง

สำหรับใน Java Script เรา尼ยมใช้เครื่องหมาย Semicolon (;) เป็นตัวกำหนด
จุดสิ้นสุดของแต่ละ Statement แม้จะไม่ใช้ข้อบังคับก็ตาม เช่น โค้ดดังไป ซึ่งแต่ละบรรทัดคือ 1

Statement โดยจะเขียนแบบคอลัมน์ทางซ้ายหรือขวาได้ แต่ส่วนใหญ่เราจะให้มี ; ปิดท้ายเหมือนกับคอลัมน์ทางซ้าย

```
var a = "ECMAScript";
var b = 8;
document.write(a + b0);
```

```
var a = "ECMAScript"
var b = 8
document.write(a + b0)
```

หากเราใช้เครื่องหมาย ; กำหนดค่าดูสิ้นสุด Statement ก็สามารถนำหลายๆ Statement มาเขียนแบบต่อเนื่องในบรรทัดเดียวกันได้ ดังโค้ดด้านไป (ส่วนใหญ่ไม่นิยมเขียนแบบนี้ เพราะจะอ่านลำบาก)

```
var a = "ECMAScript"; var b = 8; document.write(a + b0);
```

ส่วน Compound Statement ซึ่งเกิดจาก Statement ย่อยๆ รวมกัน ก็จะต้องเขียน Statement ย่อยๆ ไว้ในบล็อกที่กำหนดด้วยวงเล็บ {} เช่น โค้ดต่อไปนี้ Compound Statement

```
If(a == 0) {
    Document.write("เกิดข้อผิดพลาด");
    Return;
}
```

2.6 PHP Basic and MySQL

PHP เป็นโปรแกรมภาษาที่ทำงานในลักษณะ ภาษาสคริปต์ฝั่งเซิร์ฟเวอร์ (Server-side scripting language) คือมีลักษณะของการประมวลผลที่ฝั่งเซิร์ฟเวอร์ หลังจากนั้นก็จะทำการส่งผลลัพธ์ในรูปแบบ HTML กลับไปยังเบราว์เซอร์ ซึ่งลักษณะเช่นนี้จะทำให้เราสามารถใช้ภาษา PHP พัฒนาระบบงานในลักษณะ Dynamic Programming ได้

ไฟล์ PHP จะเหมือนกับเอกสาร HTML ทั่วไป เนื่องจากเราสามารถเขียนแท็กซึ่งเป็นคำสั่งภาษา PHP ลงไปในแท็กของ HTML ได้ ไฟล์นามสกุลของ PHP คือ .php ตัวอย่างต่อไปนี้จะแสดงถึงลักษณะของการเขียนโปรแกรมภาษา PHP โดยทำการเขียนคำสั่ง PHP ให้ทำการพิมพ์ข้อความว่า Hello World และทำการแทรก PHP ไว้ในไฟล์ HTML

```

1 <html>
2 <body>
3 <?
4 echo "Hello World";
5 ?
6 </body>
7 </html>

```

รูปที่ 2.32 ภาษา PHP แทรกใน html

ในการกลับกันเรารสามารถเปลี่ยนโปรแกรมภาษา HTML แทรกไว้ภายในภาษา PHP ได้เช่นกัน ดังตัวอย่างต่อไปนี้จะทำการพิมพ์ข้อความ Hello World เป็นตัวหนังสือสีแดง โดยใช้แท็กของ HTML เพื่อแสดงสีของตัวอักษร

```

1 <?
2 echo "<html><body>";
3 echo "<font color=red>" ;
4 echo "Hello World";
5 echo "</font>" ;
6 echo "</body></html>";
7 ?
8

```

รูปที่ 2.33 ภาษา html แทรกใน PHP

2.6.1 รูปแบบบล็อกคำสั่งของ PHP

สำหรับในโปรแกรมภาษา PHP ก็มีแท็กที่ใช้สำหรับการกำหนดจุดเริ่มต้นและจุดสิ้นสุด ของคำสั่ง PHP เช่นเดียวกัน ซึ่งในภาษา PHP มีรูปแบบบล็อกคำสั่ง 2 รูปแบบ คือ

1) ใช้ <?php...?>

```
<?php
    print "Hello World";
?>
```

2) ใช้ <script language='php'>...</script>

```
<script language='php'>
    print "Hello World";
</script>
```

เราสามารถใช้คำสั่งของ PHP รูปแบบใดก็ได้ ซึ่งจะมีผลลัพธ์ไม่แตกต่างกัน โดยจากตัวอย่างข้างต้นโปรแกรมภาษา PHP จะทำการพิมพ์ข้อความว่า Hello World ออกมาระบบลงบนบราวเซอร์

2.6.2 คำสั่งแสดงผลลัพธ์

ในการแสดงผลลัพธ์ของโปรแกรมภาษา PHP ได้จัดเตรียมคำสั่งไว้ 3 รูปแบบ คือ

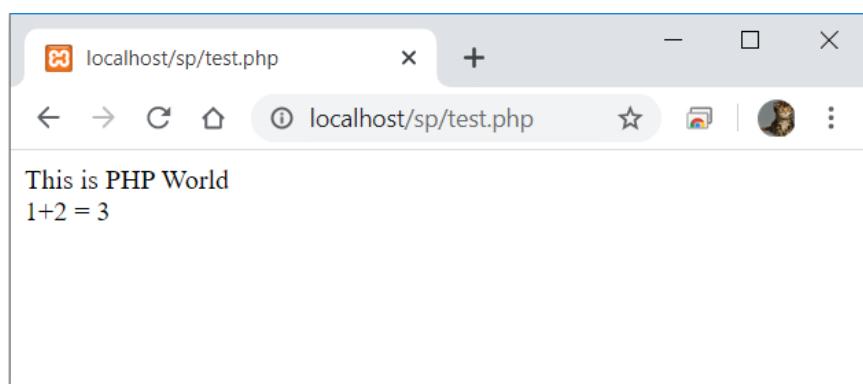
- 1) echo รูปแบบของคำสั่งนี้เป็นดังนี้

```
echo "...ข้อมูลที่ต้องการแสดงผลลัพธ์..."
```

พิจารณาด้วยตัวอย่างของการใช้คำสั่ง echo เพื่อแสดงผลลัพธ์ต่อไปนี้

```
<html>
<body>
<?php
    echo "This is PHP World";
?
<br>
<?php
    echo "1+2 = ",1+2 ;
?
</body>
</html>
```

จากโปรแกรมจะแสดงผลลัพธ์บนบราวเซอร์ดังนี้



รูปที่ 2.34 บราวเซอร์แสดงผลลัพธ์ของคำสั่ง echo

2) print รูปแบบของคำสั่งนี้เป็นดังนี้

print "...ข้อมูลที่ต้องการแสดงผลลัพธ์..."

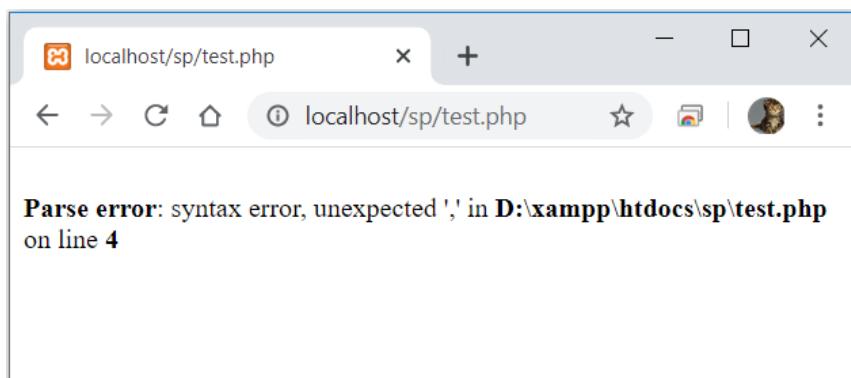
รูปแบบคำสั่งการใช้งานของคำสั่ง print และ echo จะมีรูปแบบเหมือนกัน พิจารณาตัวอย่างของการใช้คำสั่ง print เพื่อแสดงผลลัพธ์ต่อไปนี้

```
<?php
    print "This is PHP World"
?>
```

จากโปรแกรมนี้จะทำการพิมพ์ข้อความว่า This is PHP World ปรากฏบนบรรทัด
ข้อแตกต่างของคำสั่ง echo และ print ก็คือ คำสั่ง print ไม่สามารถแยกข้อความกับนิพจน์
หรือค่าของตัวแปรไดๆ ด้วยเครื่องหมาย , ได้ แต่คำสั่ง echo สามารถทำได้ สมมุติเราทำการเขียน
โปรแกรม ดังต่อไปนี้

```
<?php
    print "1+2 = ",1+2 ;
?>
```

เมื่อทำการรันโปรแกรมนี้บนบรรทัด จะแสดงข้อผิดพลาด ดังนี้



รูปที่ 2.35 บรรทัดจะแสดงความผิดพลาดจากคำสั่ง print

3) printf รูปแบบของคำสั่งนี้เป็นดังนี้

printf "...ข้อมูลที่ต้องการแสดงผลลัพธ์..."

รูปแบบคำสั่ง printf จะแตกต่างจากคำสั่ง echo และ print ก็จะต้องมีเครื่องหมายวงเล็บ^{เปิดและปิด} ตัวอย่างเช่น

```
<?php
    printf ("This is PHP World") ;
?>
```

นอกเหนือจากคำสั่ง printf ยังมีลักษณะพิเศษก็คือ คำสั่ง printf สามารถจัดรูปแบบการแสดงผลของข้อมูลที่เป็นตัวเลขได้ด้วย ดังต่อไปนี้

สัญลักษณ์	คำอธิบาย
%d	แสดงค่าอาเกิร์มั่นต์ในรูปของเลขฐานสิบ
%b	แสดง Integer ในรูปแบบฐานสอง
%c	แสดง Integer ในรูปของรหัส ASCII
%f	แสดง Integer ในรูปของเลขทศนิยม
%o	แสดง Integer ในรูปของเลขฐานแปด
%s	แสดงค่าอาเกิร์มั่นต์ในรูปของเลข字符串
%x	แสดง Integer ในรูปของเลขฐานสิบหก (ตัวพิมพ์เล็ก)
%X	แสดง Integer ในรูปของเลขฐานสิบหก (ตัวพิมพ์ใหญ่)

2.6.3 คำสั่งหมายเหตุ (Comment)

ในการเขียนโปรแกรมบางครั้งเราอาจต้องการเขียนข้อความอะไรมาก่อนอย่างเพื่ออธิบายว่าคำสั่นในแต่ละบรรทัดนั้นคืออะไร หรือมันทำงานอย่างไร เพื่อว่าเราจะเขียนกลับมาศึกษาโปรแกรมที่เขียนขึ้นอีกรัง และทำให้เข้าใจโปรแกรมเดิมได้อย่างรวดเร็ว โปรแกรมภาษา PHP มีเครื่องหมาย Comment ที่ใช้สำหรับการระบุข้อความในคลังในโปรแกรม โดยที่ส่วนของโปรแกรมที่ถูก Comment ไว้จะไม่ถูกแปลภาษา ดังนั้น ไม่ว่าเราจะเขียนอะไลงในส่วนที่ถูก Comment ไว้นี่ก็จะไม่มีความผิดพลาดเกิดขึ้น เครื่องหมาย Comment ของโปรแกรมภาษา PHP มีอยู่ 3 รูปแบบ คือ

1. เครื่องหมาย #

2. เครื่องหมาย //

ทั้ง 2 คำสั่งนี้ใช้ระบุ Comment ได้ทิ้ง 1 บรรทัด ดังตัวอย่าง

```
<?php
    echo "This is PHP World" ;
    # echo "Good Morning" ;
    // echo "How are you?" ;
?>
```

จากโปรแกรมนี้เมื่อรันโปรแกรมแล้วบริเวณที่ทำการแสดงข้อความว่า This is PHP World เพียงบรรทัดเดียวเท่านั้น

3. เครื่องหมาย /*...*/ เครื่องหมาย Comment นี้สามารถทำการ Comment ได้ครั้งละหลายบรรทัด ดังตัวอย่าง

```
<?php
    echo "This is PHP World";
    /* echo "Good Morning";
    echo "How are you?" */
?>
```

เข่นเดียวกันเมื่อรันโปรแกรมนี้ บรรทัดจะทำการแสดงข้อความว่า This is PHP World เพียงบรรทัดเดียวเท่านั้น แต่โปรแกรมนี้ใช้เครื่องหมาย /*...*/ ซึ่งจะทำให้ Comment ได้ทีละหลายบรรทัด

2.6.4 เรื่องของตัวแปรและค่าคงที่

ในการประมวลผลใดๆ ในคอมพิวเตอร์ต้องมีการจัดเก็บข้อมูลภายในหน่วยความจำของคอมพิวเตอร์ มีฉะนั้นแล้วการประมวลผลจะไม่สามารถเกิดขึ้นได้เลย หากผลง่ายๆ ก็อ้างมีการประมวลผลเกิดขึ้นแล้วต้องการจะนำผลลัพธ์ (Output) จากการประมวลผลนี้จัดเก็บเอาไว้เพื่อที่จะนำไปเป็นข้อมูลนำเข้า (Input) ของการประมวลผลอื่นๆ ลัดไป การจัดเก็บข้อมูลดังกล่าวในหน่วยความจำนี้เราสามารถกระทำได้โดยการใช้ตัวแปร

2.6.4.1 การประคัตัวแปร

ตัวแปรเป็นชื่อที่ตั้งขึ้นมาเพื่ออ้างอิงถึงข้อมูลต่างๆ ในหน่วยความจำ ซึ่งชื่อของตัวแปรนี้ไม่ใช่เราจะตั้งชื่ออะไรก็ได้ แต่จะมีการกำหนดกฎเกณฑ์ในการตั้งชื่อตัวแปร ดังนี้

- ชื่อตัวแปรต้องขึ้นต้นด้วยเครื่องหมาย \$ (dollar sign) เสมอ
- หลังเครื่องหมาย \$ จะตามด้วยชื่อตัวแปรโดยที่ตัวแรกของชื่อต้องเป็นตัวอักษร หรือเครื่องหมาย _ (underscore) เท่านั้น
- ตัวอักษรของชื่อตัวแปรสามารถเป็นตัวอักษร หรือตัวเลข หรือเครื่องหมาย _ (underscore) เพื่อผสมเป็นชื่อได้
- ชื่อของตัวแปรมีคุณสมบัติเป็น case-sensitive คือตัวอักษรตัวใหญ่และตัวเล็กถือว่าเป็นคนละตัวกัน

2.6.4.2 ชนิดของข้อมูล

ตามที่ได้กล่าวในหัวข้อที่ผ่านมาเกี่ยวกับเรื่องของตัวแปรว่าสามารถนำมาจัดเก็บข้อมูลได้ 5 ประเภท นั่นคือ ชนิดของข้อมูลที่ตัวแปรสามารถจัดเก็บได้ให้ชัดเจน โดยสามารถแบ่งชนิดของข้อมูลได้เป็น 4 ชนิด คือ

1. ข้อมูลชนิดจำนวนเต็ม (Integer)
2. ข้อมูลชนิดจำนวนทศนิยม (Floating-Point Number)
3. ข้อมูลชนิดสตริง (String)
4. ข้อมูลชนิดตรรกะ (Boolean)

2.6.4.3 ตัวแปรอาร์เรย์

ความหมายของตัวแปรอาร์เรย์ ก็คือตัวแปรที่ใช้ชื่อตัวแปรเพียงชื่อเดียว แต่มีการจัดแบ่งพื้นที่การจัดการเก็บข้อมูลในหน่วยความจำออกเป็นช่องๆ แต่ละช่องของอาร์เรย์จะมี index หรือหมายเลขที่ใช้ในการอ้างอิงถึงตำแหน่งที่อยู่ของช่องในการจัดเก็บนั่นเอง ตัวแปรอาร์เรย์ในภาษา PHP มี 3 รูปแบบดังนี้

รูปแบบที่ 1

ชื่อตัวแปรอาร์เรย์[index] = ค่าของอาร์เรย์;

รูปแบบที่ 2

ชื่อตัวแปรอาร์เรย์ = array("ค่าของอาร์เรย์ช่องที่ 0", "ค่าของอาร์เรย์ช่องที่ 1", ...)

รูปแบบที่ 3

ชื่อตัวแปรอาร์เรย์ = array("index" => "ค่าของอาร์เรย์", ...)

2.6.4.4 พังก์ชันที่เกี่ยวข้องกับตัวแปร

โปรแกรมภาษา PHP ได้จัดเตรียมพังก์ชันที่เกี่ยวข้องกับตัวแปรให้ใช้งานด้วยกันหลายพังก์ชัน ซึ่งเราไม่ต้องทำการเขียนโปรแกรมใดๆ เลย เพียงแค่รู้จักกลไกของการประมวลผลของพังก์ชันนั้นๆ และเรียกใช้งานให้ถูกต้องตามรูปแบบเท่านั้น

- พังก์ชัน gettype (ตัวแปร) เป็นพังก์ชันที่ใช้ตรวจสอบชนิดข้อมูลของตัวแปร ซึ่งอาจจะคืนค่าชนิดของตัวแปรต่างๆ
- พังก์ชัน empty (ตัวแปร) เป็นพังก์ชันที่ใช้ตรวจสอบตัวแปรว่าเก็บข้อมูลไว้หรือไม่ โดยถ้าตัวแปรเก็บข้อมูลไว้จะคืนค่าเท็จ (false) ซึ่งค่าเท็จนี้จะเป็นช่องว่าง แต่ถ้าไม่ได้เก็บข้อมูลไว้จะคืนค่าจริง (true) ซึ่งค่าจริงนี้จะเป็นหมายเลข 1 ไว้

- พังก์ชัน isset (ตัวแปร) พังก์ชันนี้ใช้สำหรับตรวจสอบว่ามีตัวแปรตามชื่อที่ระบุไว้หรือไม่ โดยถ้ามีตัวแปรนั้นจะคืนค่าจริง (true) แต่ถ้าไม่มีจะคืนค่าเท็จ (false)
- พังก์ชัน unset (ตัวแปร) พังก์ชันนี้ใช้ทำลายตัวแปรและทำการคืนพื้นที่ในหน่วยความจำ ดังนั้นหลังจากการใช้พังก์ชันนี้กับตัวแปรใดๆ แล้วตัวแปรนั้นก็จะถือว่าไม่มีตัวตนคือไม่มีตัวแปรนั้นอีกแล้ว

2.6.4.5 ค่าคงที่

ค่าคงที่ (Constant) คือชื่อที่กำหนดขึ้นมาเพื่อเก็บค่าใดๆ เอาไว้ โดยปกติเราจะนำค่าคงที่มาเก็บค่าใดๆ ที่ไม่ค่อยมีการเปลี่ยนแปลง รูปแบบการใช้งานค่าสั่งค่าคงที่ คือ

Define (ชื่อค่าคงที่, ค่า)

2.6.5 การสร้างฟังก์ชันของ PHP

ก่อนที่เราจะใช้งานฟังก์ชันใดๆ เราจะต้องสร้างฟังก์ชันขึ้นก่อน หลังจากนั้นค่อยเรียกชื่อฟังก์ชันนั้นเมื่อต้องการใช้งาน ซึ่งการสร้างฟังก์ชันนี้เราอาจจะเรียกว่าเป็นการประกาศฟังก์ชัน โดยรูปแบบของการประกาศฟังก์ชันในภาษา PHP เป็นดังนี้

```
Function ชื่อฟังก์ชัน([argument1, argyment2, ...])
{
    Statement;
}
```

ตารางที่ 2.1 รายละเอียดของการประกาศฟังก์ชัน PHP

ชื่อฟังก์ชัน	เป็นชื่อที่ตั้งขึ้นเพื่อเรียกใช้งานฟังก์ชัน
argument1, argument2, ...	เป็นตัวแปรที่ตั้งขึ้นมา เพื่อรับค่าจากการที่ฟังก์ชันส่งผ่านค่าเข้ามาประมวลผลภายในฟังก์ชัน โดยจะมีก่อาร์กิวเมนต์ได้แต่ถ้าในการใช้งานฟังก์ชันไม่มีการส่งผ่านค่าในส่วนนี้ก็ไม่ต้องระบุอาร์กิวเมนต์ เพียงวงเล็บเปิด-ปิดไว้เท่านั้น
Statement	เป็นคำสั่งที่ประมวลผลในฟังก์ชัน

ยกตัวอย่างการประกาศฟังก์ชันชื่อว่า asterisk โดยเป็นฟังก์ชันที่ใช้พิมพ์เครื่องหมายคอกขัน จำนวน 30 ครั้ง รายละเอียดของฟังก์ชันนี้เป็นดังนี้

```

1 <html>
2 <body>
3 <?php
4     function asterisk()
5     {
6         /*เรียกฟังก์ชัน*/
7         echo "*****<br>";
8     }
9     /*เรียกฟังก์ชัน*/
10    asterisk();
11    echo "My name is Sarut <br>";
12    /*เรียกฟังก์ชัน*/
13    asterisk();
14 ?>
15 </body>
16 </html>

```

Ln:16 Col:8 Sel:0|0

รูปที่ 2.36 การสร้าง และเรียกใช้ function ของภาษา php

2.6.6 พังก์ชันของ PHP

เราได้รู้การสร้างพังก์ชันไว้ใช้งาน ซึ่งผู้พัฒนาจะต้องเขียนโปรแกรมขึ้นเองทั้งหมด และต้องอาศัยความรู้ในการประยุกต์ใช้พังก์ชัน การเรียกพังก์ชันใช้งาน แต่เพื่อความสะดวกของโปรแกรม PHP ได้จัดเตรียมพังก์ชันสำหรับใช้บ่อยๆ ไว้ใช้งานได้โดยที่เราไม่ต้องเขียนโปรแกรมเลย เพียงแต่รู้จักชื่อพังก์ชัน และการเรียกใช้งานพังก์ชันให้ถูกต้อง พังก์ชันในลักษณะนี้มีชื่อว่า Pre-defined-Function

ตารางที่ 2.2 พังก์ชันเกี่ยวกับสตริง PHP

ชื่อพังก์ชัน	รายละเอียด
addslashes()	ใช้เพิ่มเครื่องหมาย \ ไว้ข้างหน้าให้กับตัวอักษรพิเศษ
chr() และ ord()	ใช้จัดการรหัสແອສกີของตัวอักษร
explode() และ implode()	ใช้แบ่งสตริงออกเป็นสตริงย่อยๆ และรวมสตริง
htmlspecialchars()	ใช้แปลงตัวอักษรพิเศษ
nl2br()	ใช้แทรกแท็ก หรือแท็ก
strcmp()	ใช้เปรียบเทียบสตริง
strlen()	ใช้หาความยาวสตริง
strrev()	ใช้กลับลำดับของตัวอักษร
substr()	ใช้ดึงบางส่วนของสตริงออกมากจากสตริงหลัก
trim(), ltrim และ rtrim()	ใช้ตัดช่องว่าง แท็บ ตัวอักษรชี้บนบรรทัดใหม่

ตารางที่ 2.3 พังก์ชันเกี่ยวกับคณิตศาสตร์ PHP

ชื่อฟังก์ชัน	รายละเอียด
abs()	ใช้หาค่าสมบูรณ์ของตัวเลข
max() min()	หากคุณต้องการที่มากที่สุด, หากคุณต้องการที่น้อยที่สุด
pow()	ใช้หาค่าเลขยกกำลัง
round()	ใช้ปัดเลขทศนิยม
sqrt()	ใช้หาค่าที่สองของตัวเลขได้ๆ

ตารางที่ 2.4 พังก์ชันเกี่ยวกับวันเวลา PHP

ชื่อฟังก์ชัน	รายละเอียด
checkdate()	ใช้ตรวจสอบวันเดือนปี
date()	ใช้ดึงวันเดือนปี และเวลาปัจจุบัน
getdate()	ให้ผลลัพธ์เป็นวันเวลาปัจจุบัน
time()	จะคืนค่าวัน/เวลาปัจจุบันของเครื่อง

2.6.7 Session และ Cookie

เซสชัน (Session) มีคุณสมบัติใช้ส่งค่าของตัวแปรที่สร้างขึ้นจากไฟล์หนึ่งไปแสดงผลยังอีกไฟล์ได้ ทำให้เราสามารถทำการติดตามและตรวจสอบผู้ใช้ได้ การใช้งานเซสชันในภาษา PHP ได้จัดเตรียมฟังก์ชันให้ใช้งานอยู่หลายฟังก์ชัน เช่น Session_start() และ Session_destroy()

คุกกี้ (Cookies) คือข้อมูลที่เราส่งไปเก็บไว้ในเครื่องของผู้ใช้ เพื่อประยุกต์ในการตรวจสอบ และติดตามการทำงานของผู้ใช้ โดยการทำงานบนเว็บเซอร์ฟเวอร์จะทำการส่งข้อมูลของผู้ใช้ที่จัดเก็บไว้ในเครื่องมายังเว็บเซอร์ฟเวอร์ และจะนำคุกกี้ดังกล่าวมาเก็บไว้ในตัวแปรทำให้สามารถตรวจสอบค่าต่างๆ ของผู้ใช้ได้ ฟังก์ชันที่สำคัญในการจัดการคุกกี้ คือ setcookie(), ob_start(), ob_end_flush()

2.6.8 ฐานข้อมูล MySQL

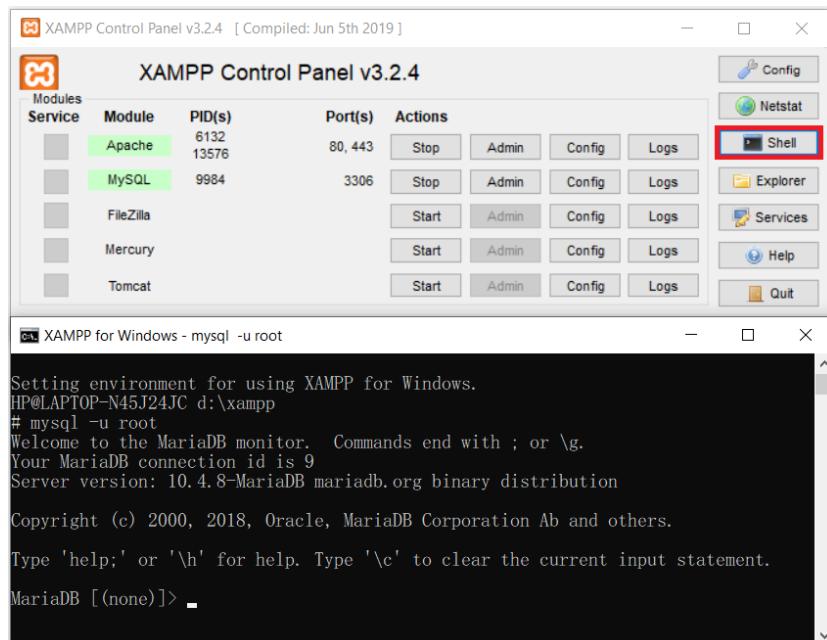
ฐานข้อมูลเป็นองค์ประกอบหนึ่งที่สำคัญในการพัฒนาแอ��เพลิเคชัน (Application) เนื่องจากระบบงานต่างๆ จะมีการจัดเก็บข้อมูลไว้ในฐานข้อมูล เพื่อให้สามารถเข้าถึงข้อมูลที่ฐานข้อมูลได้ คำว่า “เข้าถึงข้อมูล” ในที่นี่คือ การที่ระบบสามารถเพิ่ม, ลบ, แก้ไขข้อมูล และจัดทำรายงานได้ ซึ่งถือว่าเป็นฟังก์ชันพื้นฐานของระบบต่างๆ เช่น ระบบงานทางธุรกิจ เราจัดเก็บข้อมูล

ของสินค้าคงคลังเพื่อเก็บรายละเอียดสินค้าและสต็อกสินค้าของบริษัท ถ้ามีลูกค้ามาสั่งซื้อสินค้า เรา
ก็สามารถทำการตรวจสอบว่ามีสินค้าที่ลูกค้าต้องการหรือไม่ หลังจากนั้นเมื่อลูกค้าซื้อสินค้าแล้ว
เราจะสามารถทำการแก้ไขหรือปรับปรุงจำนวนสินค้าที่ขายไปได้ เป็นต้น

2.6.8.1 การใช้งาน MySQL

หลังจากทำการติดตั้งโปรแกรม XAMPP ในหัวข้อที่ 2.2 แล้ว เราสามารถเข้าไปใช้งานฐานข้อมูล MySQL ได้ดังนี้

1. เปิดส่วนควบคุมของ XAMPP ขึ้นมา โดยไปที่ Start>All Program>XAMPP>XAMPP Control Panel
2. คลิกที่ปุ่ม shell จากนั้นพิมพ์คำสั่ง mysql -u root ! เพื่อเรียกใช้งาน MySQL
3. จะเข้าสู่หน้าจอการใช้งาน MySQL



รูปที่ 2.37 การเรียกใช้งานฐานข้อมูล MySQL ใน XAMPP

ในการใช้งานฐานข้อมูล MySQL จำเป็นต้องรู้จักคำสั่งพื้นฐานในการเข้าไปดูรายละเอียดของฐานข้อมูล ดังตารางสรุปต่อไปนี้

ตารางที่ 2.5 คำสั่งเพื่อดำเนินการกับฐานข้อมูล

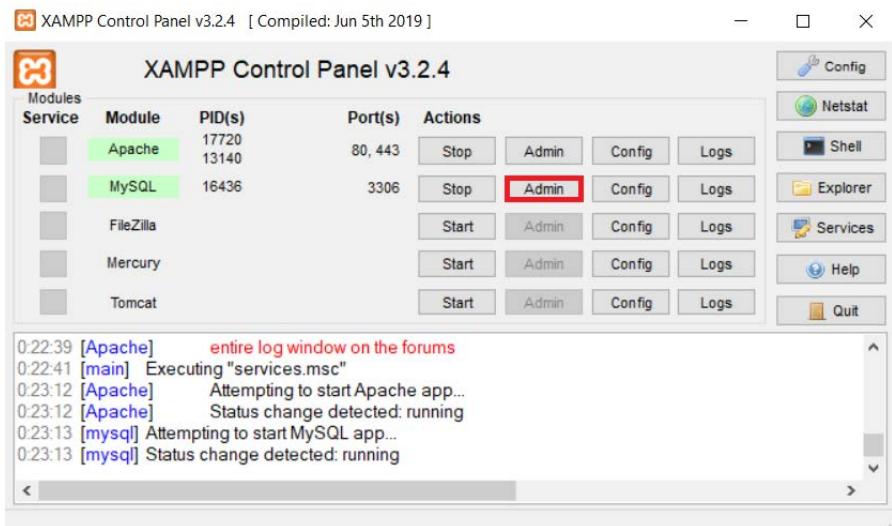
คำสั่ง	คำอธิบาย
Show database;	แสดงรายชื่อฐานข้อมูลทั้งหมด
Use ชื่อฐานข้อมูล;	เลือกฐานข้อมูลที่จะทำงานด้วย
Show table;	แสดงรายชื่อตารางทั้งหมดภายในฐานข้อมูล
Show column from ชื่อตาราง;	แสดงชื่อฟิลด์และรายละเอียดของตาราง
คำสั่ง	คำอธิบาย
Create Database ชื่อฐานข้อมูล;	สร้างฐานข้อมูลใหม่
Create Table (ชื่อฟิลด์ และรายละเอียดต่างๆ);	สร้างตารางข้อมูลใหม่
Insert into ชื่อตาราง (ชื่อฟิลด์) values (ค่าของฟิลด์);	เพิ่มข้อมูลเรคอร์ดใหม่ลงในตาราง
Select ชื่อฟิลด์ From ชื่อตาราง [Where เงื่อนไข];	ดึงข้อมูลจากตารางที่กำหนดและตามเงื่อนไข
Update ชื่อตาราง Set ชื่อฟิลด์ = ค่าที่แก้ไข [Where เงื่อนไข];	แก้ไขข้อมูลในตารางตามเงื่อนไขที่ต้องการ
Delete Form ชื่อตาราง Where เงื่อนไข;	ลบเรคอร์ดจากตารางที่ระบุ และตามเงื่อนไข
exit	ออกจากการทำงาน MySQL

2.6.9 การจัดการฐานข้อมูล MySQL ด้วย PhpMyAdmin

จากเนื้อหาในหัวข้อที่ผ่านมาเราได้เรียนรู้การใช้งานฐานข้อมูล MySQL รวมทั้งเรียนรู้คำสั่งที่ใช้ในการจัดการฐานข้อมูลด้วย เราจะเห็นข้อเสียของการใช้งานดังกล่าวคือ ความยากลำบากในการจัดการฐานข้อมูล เช่น การที่ต้องเข้าใจและเรียกใช้คำสั่งได้ถูกต้อง มีขั้นตอนแล้วเราจะไม่สามารถใช้งาน MySQL ได้เลย ในปัจจุบันเราสามารถแก้ปัญหาความยากลำบากดังกล่าวได้โดยการใช้โปรแกรมสำหรับที่มีคุณสมบัติของการทำงานในลักษณะ Graphic User Interface (GUI) โดยที่ไม่ต้องเรียนรู้คำสั่งในการจัดการฐานข้อมูลเลย ผู้ใช้เพียงคลิกเลือกคำสั่งต่างๆ ที่จัดเตรียมมาให้แล้ว เราจะสามารถใช้งานฐานข้อมูล MySQL ได้อย่างง่ายได้ โปรแกรมที่ใช้จัดการฐานข้อมูลดังกล่าวมีอยู่หลายโปรแกรมแต่ที่นิยมใช้มากที่สุดก็คือ โปรแกรม PhpMyAdmin

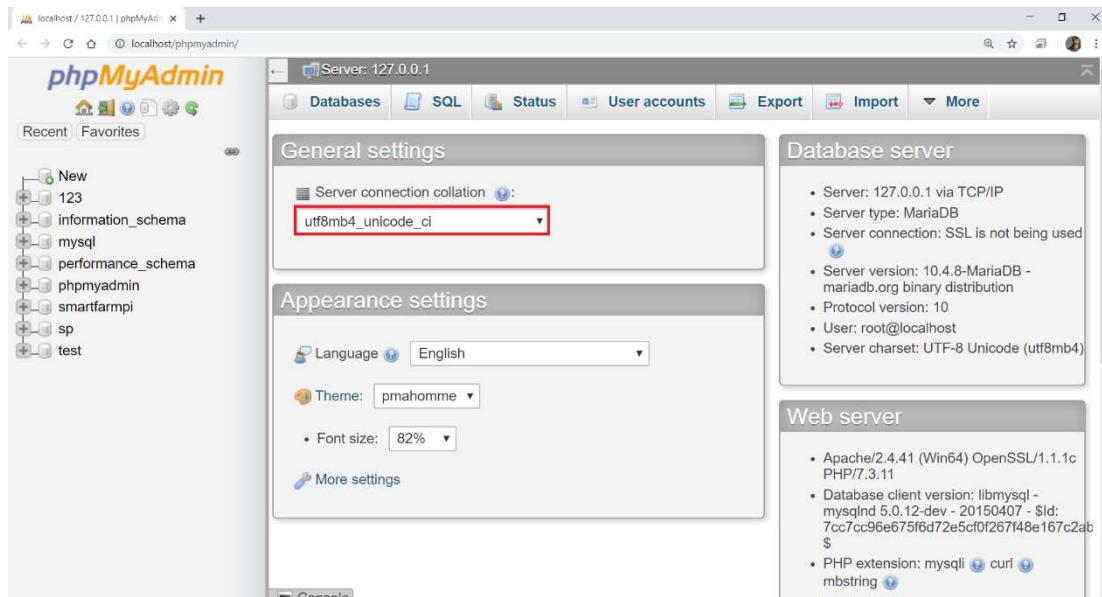
2.6.9.1 การใช้งานโปรแกรม PhpMyAdmin

การเปิดใช้งานโปรแกรม PhpMyAdmin ให้ทำการเปิด XAMPP Control Panel แล้วคลิกที่ปุ่ม Admin ของ MySQL ดังนี้



รูปที่ 2.38 การเปิดใช้งานโปรแกรม PhpMyAdmin

หน้าหลักของ PhpMyAdmin แบ่งพื้นที่ของหน้าเว็บออกเป็น 2 ส่วน คือพื้นที่สำหรับจัดการฐานข้อมูลที่ต้องการจัดการ นอกจากนั้นรายชื่อตาราง (Table) ก็จะถูกกำหนดไว้ในส่วนนี้ด้วย พื้นที่สำหรับจัดการรายละเอียดต่างๆ ของฐานข้อมูลและตารางที่เราสามารถจัดการอยู่ ก่อนเริ่มต้นให้ตั้งค่ารหัสอักขระที่ใช้ติดต่อกันฐานข้อมูล (Collation) เป็น utf8mb4_unicode_ci หรือ utf8_unicode_ci เพื่อให้เก็บข้อมูลเป็นภาษาไทยได้อย่างถูกต้อง

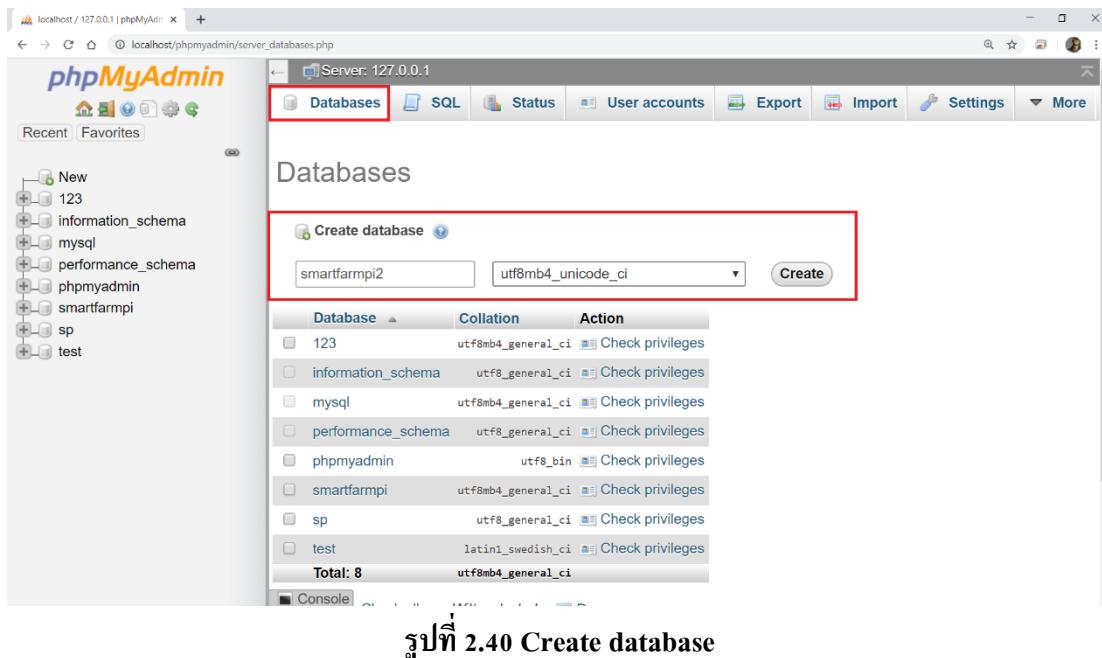


รูปที่ 2.39 กำหนดรหัสอักขระ utf8mb4_unicode_ci หรือ utf8_unicode_ci

การสร้างและลบฐานข้อมูล

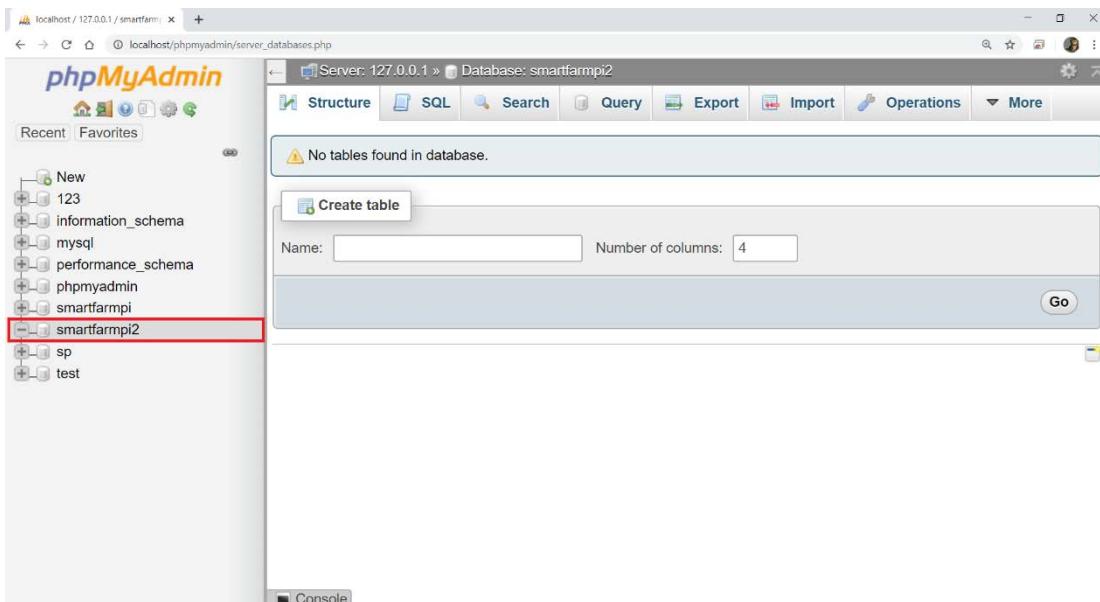
ทดลองสร้างฐานข้อมูลชื่อว่า SmartfarmPi2 ดังนี้

- จากหน้าหลักของ PhpMyAdmin คลิกที่ลิ้งก์ Database บนแนวโน้มน้ำร้าว จากนั้นพิมพ์ชื่อ SmartfarmPi2 ลงไว้ในช่อง Create database หลังจากนั้นให้คลิกที่ปุ่ม Create



รูปที่ 2.40 Create database

จะปรากฏหน้าจอแสดงว่าสร้างฐานข้อมูลสำเร็จ และแสดงฐานข้อมูล smartfarmpi2 ที่ค้านช้ายังคงรูป



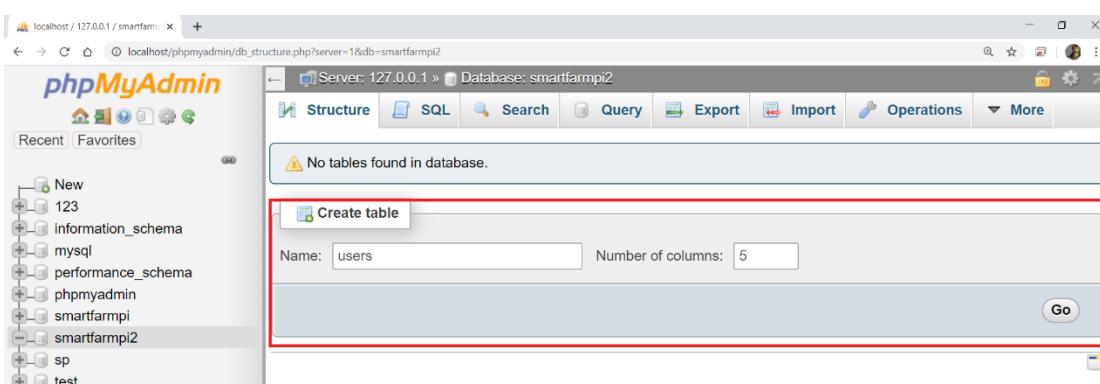
รูปที่ 2.41 หน้าจอหลังจากสร้างฐานข้อมูล

ให้คลิกที่ smartfarmpi2 ด้านซ้ายเพื่อเปิดฐานข้อมูลขึ้นมา จะพบข้อความว่า No table found in database. ซึ่งหมายความว่าขณะนี้ในฐานข้อมูลนี้ไม่มีตารางเลข โดยวิธีการสร้างตารางภายในฐานข้อมูลจะได้ก่อตัวถึงในส่วนถัดไป

การสร้างตาราง

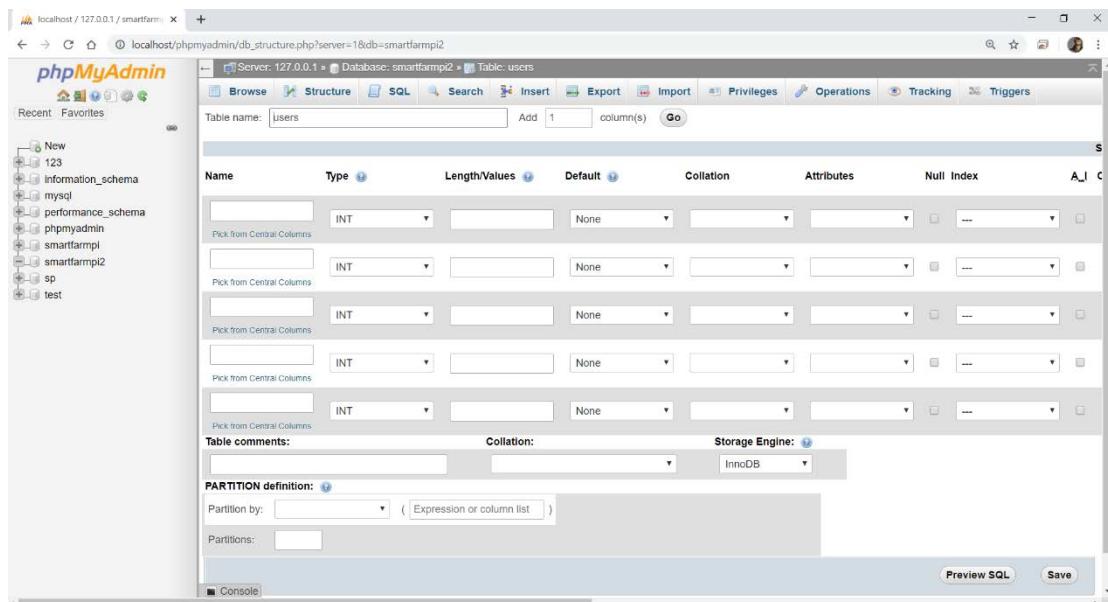
ทดลองสร้างตารางชื่อว่า users โดยมีฟิลด์ทั้งหมด 5 ฟิลด์ ดังนี้

1. พิมพ์ชื่อตาราง users ในช่อง Name และพิมพ์จำนวนฟิลด์เท่ากับ 5 ในช่อง Number of field หลังจากนั้นให้คลิกที่ปุ่ม Go ดังนี้



รูปที่ 2.42 การตั้งชื่อตารางและกำหนดจำนวนฟิลด์

2. จะปรากฏหน้าจอในการกำหนดรายละเอียดของตาราง ดังนี้



รูปที่ 2.43 หน้าจอการกำหนดรายละเอียดของตาราง

จากรูป จะปรากฏหน้าจอให้กรอกรายละเอียดต่างๆ ของตาราง (ในที่นี้มีทั้งหมด 5 ค่า โดย 1 ค่า เท่ากับ 1 ฟิลด์) โดยประกอบไปด้วยรายการข้อมูล ดังนี้

- Name

ใช้สำหรับพิมพ์ชื่อฟิลด์

- Type

ใช้สำหรับกำหนดชนิดของข้อมูลที่ใช้ในฟิลด์นั้นๆ

- Length/Values

ใช้สำหรับกำหนดความยาวของฟิลด์นั้นๆ

- Default

ใช้สำหรับกำหนดค่าของฟิลด์ในกรณีที่ไม่ได้มีการกรอกข้อมูลใดๆ

- Attributes

ใช้สำหรับกำหนดลักษณะของฟิลด์นั้นๆ

- Collation

ใช้สำหรับการกำหนดการเรียงลำดับของข้อมูลภายในตาราง

- Null

ใช้สำหรับกำหนดค่าฟิลด์นั้นๆ จะให้มีค่าว่างเปล่าได้หรือไม่

- Index

ใช้สำหรับกำหนดให้ฟิลด์นั้นเป็นคีย์หลัก (Primary key)

- A_I

ใช้สำหรับกำหนดให้ฟิลด์นั้นมีลักษณะที่เป็นแบบ auto_increment

จากรายการข้อมูลเราไม่จำเป็นต้องกรอกข้อมูลให้ครบถ้วนรายการก็ได้ ในที่นี้
สมมุติว่าต้องการสร้างตาราง users ที่มีรายละเอียดต่างๆ ดังนี้

ชื่อฟิลด์	ชนิดข้อมูล	ความยาว	อื่นๆ
user_id	INT	10	ให้เป็นคีย์หลัก (ไพร์มารี) และเป็น auto_increment
username	VARCHAR	255	
password	VARCHAR	64	
email	VARCHAR	100	
user_type	VARCHAR	16	

ดังนั้น เราจะสามารถกำหนดรายละเอียดของตาราง product นี้ด้วยโปรแกรม

PhpMyAdmin ได้ดังนี้

The screenshot shows the PhpMyAdmin interface with the 'users' table selected. The table structure is defined as follows:

- Name:** user_id (Type: INT, Length/Values: 10, Default: None, Collation: None, Attributes: PRIMARY, Null: No, Index: Yes)
- Name:** username (Type: VARCHAR, Length/Values: 255, Default: None, Collation: None, Attributes: None, Null: Yes, Index: No)
- Name:** password (Type: VARCHAR, Length/Values: 64, Default: None, Collation: None, Attributes: None, Null: Yes, Index: No)
- Name:** email (Type: VARCHAR, Length/Values: 100, Default: None, Collation: None, Attributes: None, Null: Yes, Index: No)
- Name:** user_type (Type: VARCHAR, Length/Values: 16, Default: None, Collation: None, Attributes: None, Null: Yes, Index: No)

รูปที่ 2.44 กำหนดรายละเอียดของตาราง

หลังจากกำหนดรายละเอียดต่างๆ ของตารางครบแล้วให้ทำการคลิกที่ปุ่ม Save
โปรแกรมก็จะสร้างตารางขึ้นมาเก็บไว้ในฐานข้อมูล

เราสามารถดูรายละเอียดของตารางได้ โดยคลิกที่ Structure ด้านหลังตารางที่ต้องการ ดังนี้

รูปที่ 2.45 หน้าจอสรุประยะละเอียดของตาราง

การแก้ไขรายละเอียดของตาราง

จากรูปสรุประยะละเอียดของตาราง เราสามารถทำการแก้ไขรายละเอียดของตารางได้โดยคลิกที่ Check box หน้าชื่อฟิลด์ที่ต้องการแก้ไข หรือลบออก โดยให้เลือกฟิลด์ที่ต้องการ และคลิกที่ไอคอน Change ถ้าต้องการแก้ไข หรือคลิกที่ไอコン Drop ถ้าต้องการลบฟิลด์นั้นออกจากตาราง

ยกตัวอย่างโดยทำการคลิกที่ไอคอน Change ด้านท้ายฟิลด์ที่ชื่อ user_id !เพื่อแก้ไขรายละเอียดของฟิลด์ จะปรากฏหน้าจอดังนี้

รูปที่ 2.46 ตัวอย่างการแก้ไขรายละเอียดของฟิลด์ user_id

สมมุติเราต้องการแก้ไขความยาวของฟิลด์ user_id จากความยาว 10 เป็น 5 กีทำ การแก้ไขค่าความยาวในช่อง Length/Values หลังจากนั้นก็ให้ทำการคลิกที่ปุ่ม Save ก็จะปรากฏหน้าจอดังนี้

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user_id	int(5)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	password	varchar(64)	utf8mb4_unicode_ci		No	None			Change Drop More
4	email	varchar(100)	utf8mb4_unicode_ci		No	None			Change Drop More
5	user_type	varchar(16)	utf8mb4_unicode_ci		No	None			Change Drop More

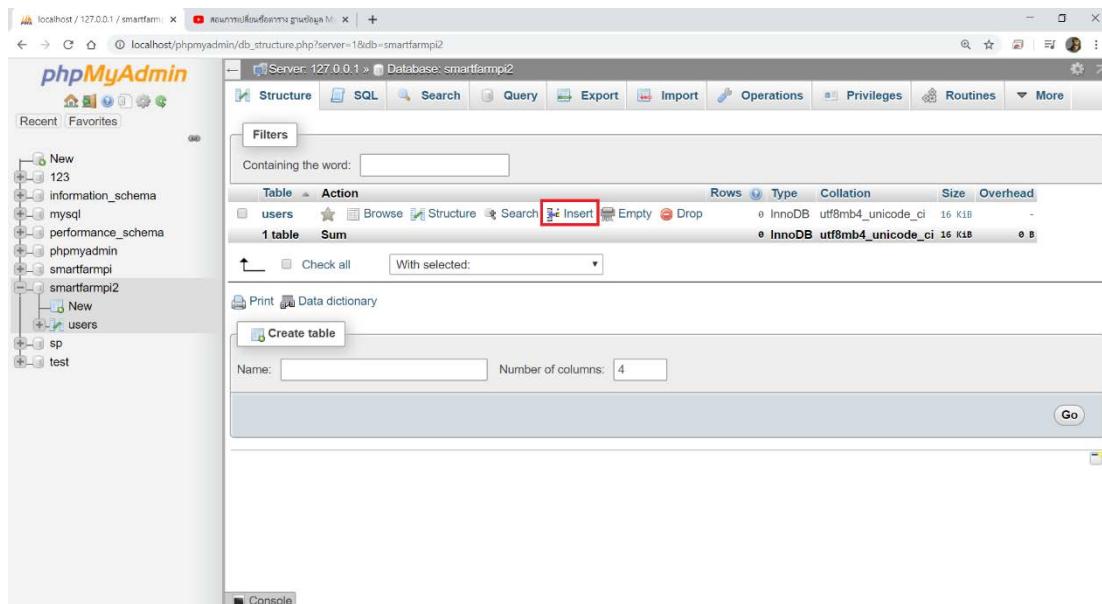
Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	user_id	0	A	No	

รูปที่ 2.47 ผลลัพธ์หลังจากการแก้ไขความยาวของฟิลด์ user_id

การเพิ่มเรคอร์ดใหม่ในตาราง

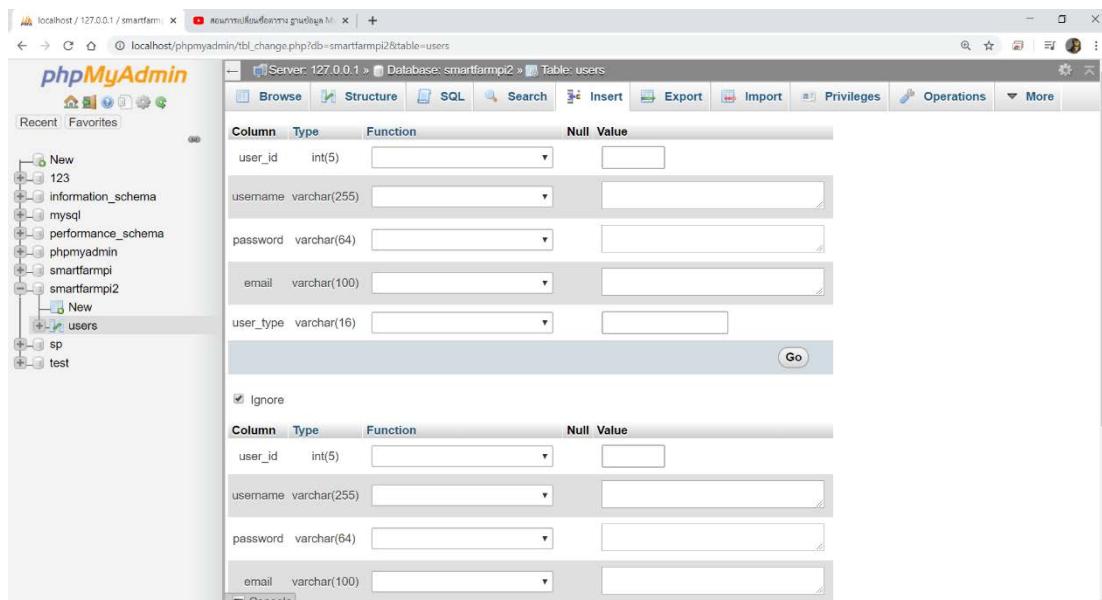
การเพิ่มเรคอร์ดใหม่ในตารางมีขั้นตอนดังนี้

1. คลิกที่ลิงก์ [Insert](#) ด้านหลังตาราง users



รูปที่ 2.48 สำหรับการเพิ่มเรคอร์ดใหม่ในตาราง

2. จะปรากฏหน้าจอการเพิ่มเรคอร์ดใหม่ในตารางดังนี้



รูปที่ 2.49 หน้าจอการเพิ่มเรคอร์ดใหม่

3. ที่คอลัมน์ “ค่า” ให้กรอกรายละเอียดลงไปในแต่ละช่องของ Text box ในที่นี่ ทดสอบใส่ค่าดังนี้

user_id : 00001, username : sarut, password : 123456789 : email :
sarut@gmail.com, user_type : admin

หลังจากนั้นคลิกที่ปุ่ม Go บริเวณด้านล่าง จะได้ผลลัพธ์ดังนี้

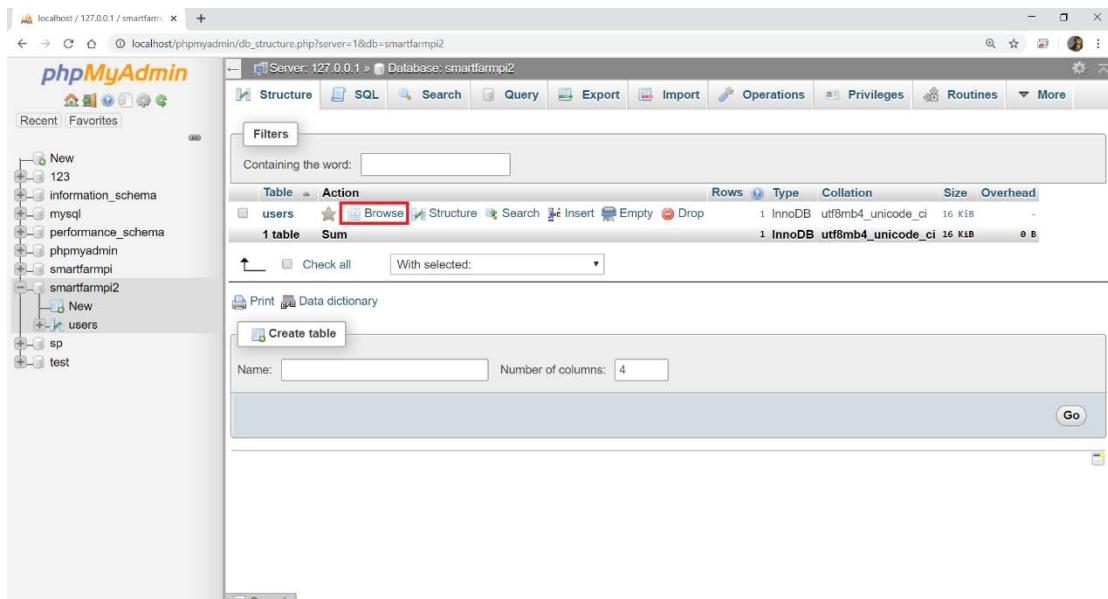
	user_id	username	password	email	user_type
	1	sarut	123456789	sarut@gmail.com	admin

รูปที่ 2.50 ผลลัพธ์การเพิ่ม rekord ใหม่

การคุ้มครองข้อมูลภายในตาราง

การคุ้มครองที่ได้รับรายละเอียดง่ายๆ แล้ว มีขั้นตอนดังนี้

1. คลิกที่ลิงก์ [Browse](#) ด้านหลังตาราง users
2. จะปรากฏหน้าจอแสดง rekord ทั้งหมดที่อยู่ภายใต้ตาราง



รูปที่ 2.51 ลิ้งค์ **Browse** สำหรับการดูข้อมูลในตาราง

การแก้ไขข้อมูลเรคอร์ด

สมมุติเราต้องการแก้ไขเรคอร์ดที่ 1 (00002, aot, 987654321, aot@gmail.com, user) ให้ทำตามขั้นตอนดังนี้

1. บริเวณด้านหน้าของแต่ละฟิลด์ให้คลิกที่ไอคอน ด้านหน้าเรคอร์ดที่ต้องการจะปรับเปลี่ยนข้อมูล

Column	Type	Function	Null	Value
user_id	int(5)		00002	
username	varchar(255)		aot	
password	varchar(64)		987654321	
email	varchar(100)		aot@gmail.com	
user_type	varchar(16)		user	

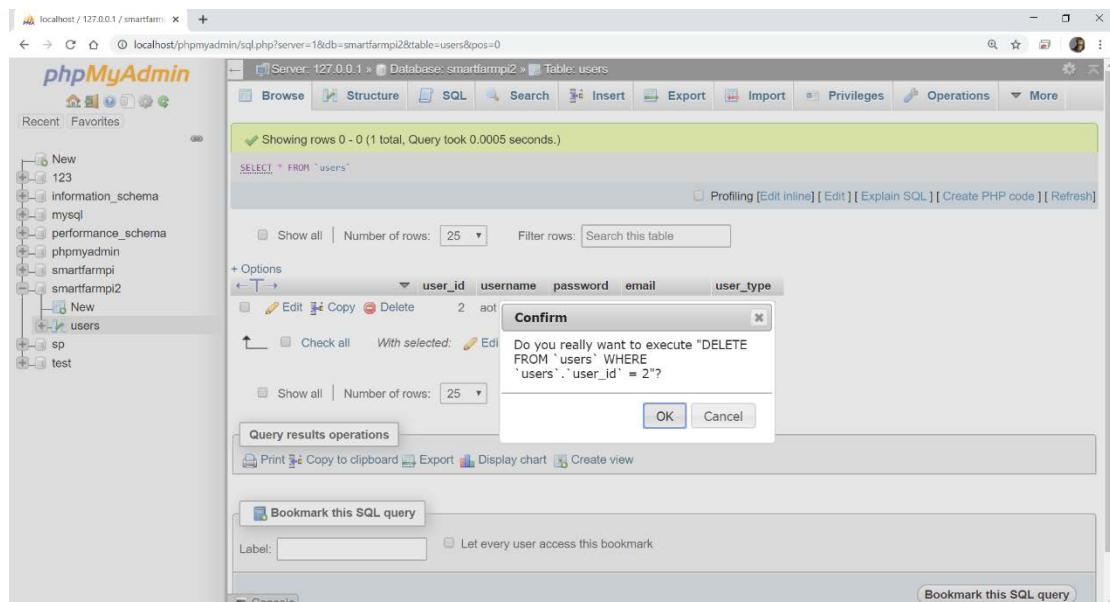
รูปที่ 2.52 หน้าของการแก้ไขเรคอร์ด

2. ทำการแก้ไขข้อมูลของเรkorดใหม่ (ที่คอลัมน์ Value) ตามที่ต้องการ หลังจากนั้นทำการคลิกที่ปุ่ม Go

การลบเรkorด

หากต้องการลบเรkorดที่ไม่ต้องการทิ้งไป ให้ทำการขันตอนดังนี้

- บริเวณด้านหน้าของแต่ละฟิลด์ให้คลิกที่ไอคอน Delete ด้านหน้าเรkorดที่ต้องการลบก็จะปรากฏหน้าจอดังนี้



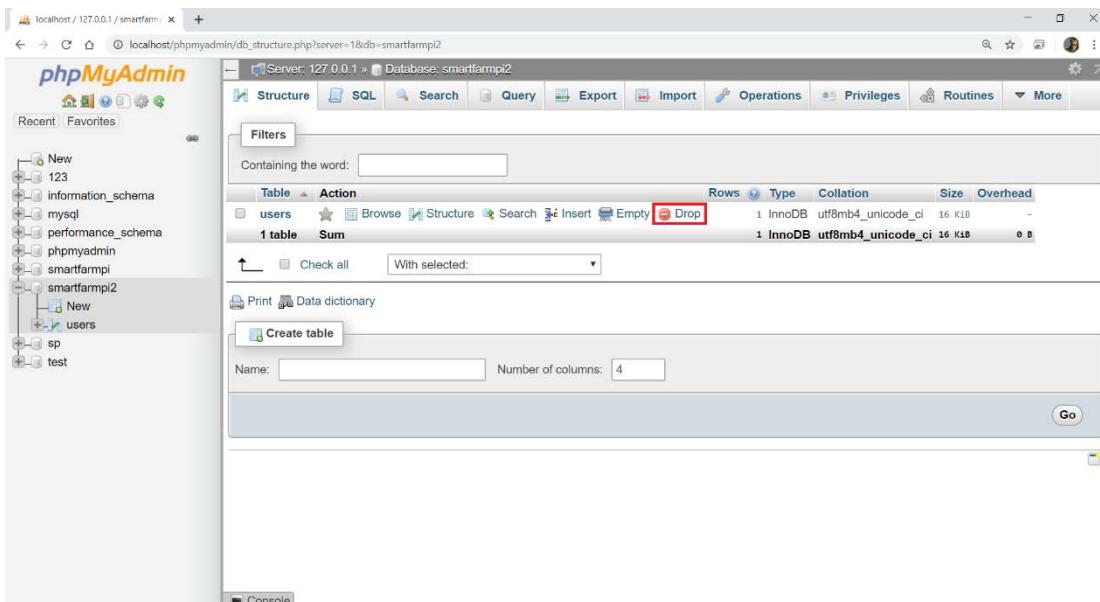
รูปที่ 2.53 หน้าจอแสดง Dialogue Box ยืนยันการลบ

- จะปรากฏหน้าจอเพื่อยืนยันการลบเรkorดนี้ โดยถ้าต้องการลบก็ให้ทำการคลิกที่ปุ่ม “OK” แต่ถ้าไม่ต้องการลบให้คลิกที่ปุ่ม “Cancel”

การลบตาราง

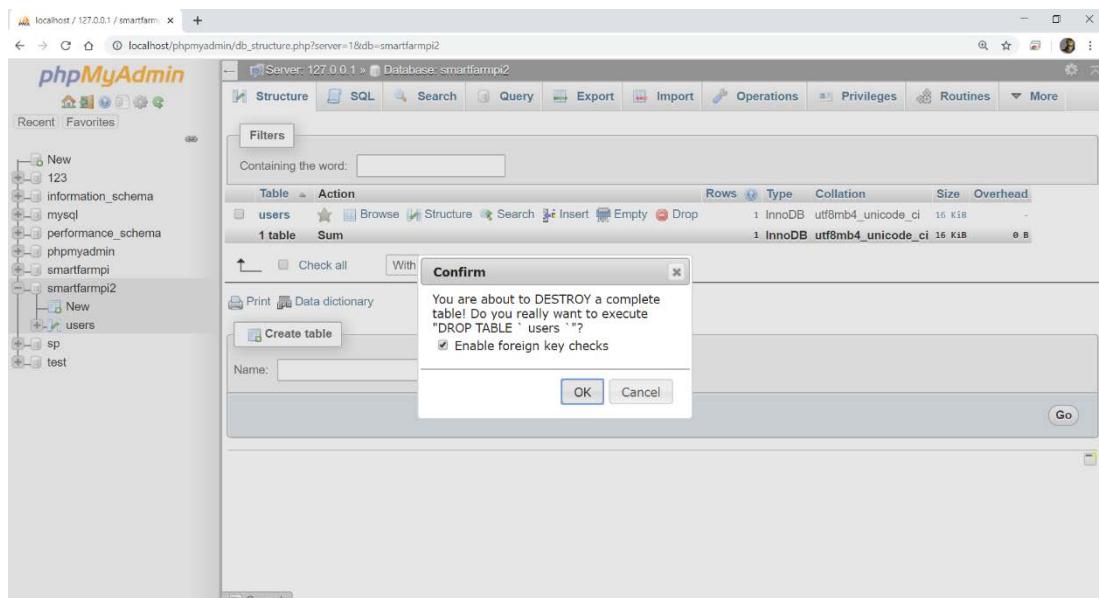
การลบตาราง มีขั้นตอนดังนี้

- คลิกที่ลิ้งค์ Drop ด้านหลังตาราง users



รูปที่ 2.54 ลิงก์ Drop สำหรับลบตาราง

2. จะปรากฏหน้าจอเพื่อให้ยืนยันการลบตารางนี้ ดังนี้



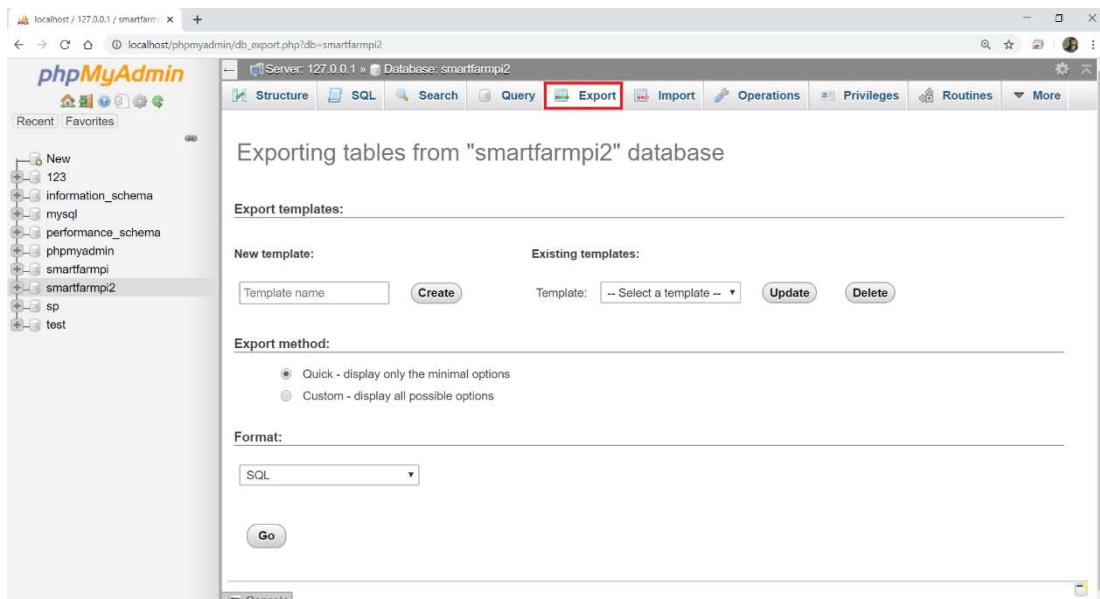
รูปที่ 2.55 หน้าจอยืนยันการลบตาราง

3. ถ้าต้องการลบตารางให้คลิกที่ปุ่ม “OK” แต่ถ้าไม่ต้องการลบให้คลิกที่ปุ่ม “Cancel”

การส่งออกฐานข้อมูล

เมื่อเราจัดสร้างฐานข้อมูลโดยใช้โปรแกรม PhpMyAdmin แล้ว และต้องการนำฐานข้อมูลดังกล่าวไปใช้กับเครื่องคอมพิวเตอร์เครื่องอื่น เราสามารถทำได้ตามขั้นตอนดังนี้

1. เลือกฐานข้อมูล smartfarmpi2 แล้วคลิกที่ลิงก์ Export บนแนวโน้มหัวข้อ



รูปที่ 2.56 ลิงก์ Export สำหรับการส่งออกฐานข้อมูล

2. คลิกที่ปุ่ม Go เพื่อบันทึกฐานข้อมูลนี้เป็นไฟล์ลงในเครื่อง ซึ่งไฟล์ที่ได้จะมีนามสกุลเป็น .sql ที่สามารถนำไปติดตั้งยังเครื่องคอมพิวเตอร์เครื่องอื่นๆ ได้

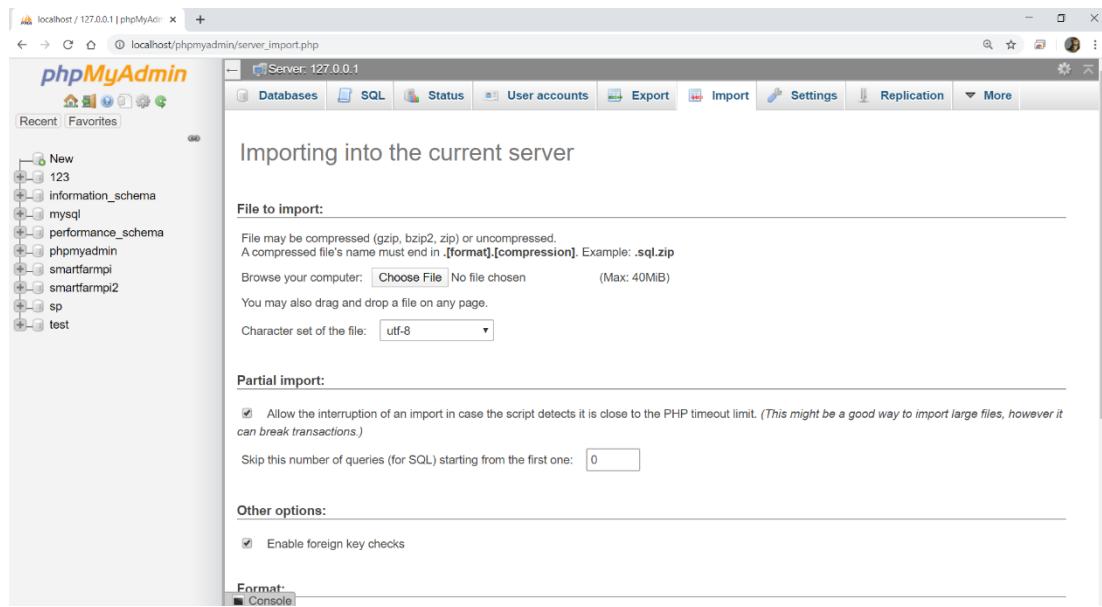
การนำเข้าฐานข้อมูล

หลังจากเราได้ทำการส่งออกฐานข้อมูลในหัวข้อที่ผ่านมาแล้ว เราจะสามารถนำไปติดตั้งที่เครื่องคอมพิวเตอร์เครื่องอื่นๆ ได้ (ถ้าไม่ทำขั้นตอนนี้และอย่างจะได้ฐานข้อมูลที่เหมือนกันบนเครื่องคอมพิวเตอร์เครื่องอื่น เราจะต้องเริ่มสร้างฐานข้อมูลใหม่ตั้งแต่ต้น) ขั้นตอนการนำเข้าฐานข้อมูลมีลำดับขั้นตอนดังนี้

1. ไปที่หน้าแรกของ PhpMyAdmin โดยคลิกที่ไอคอนรูป ที่บาร์เว็บพื้นที่ฟังชั่น
2. ทำการสร้างฐานข้อมูลเปล่าขึ้นมาเสียก่อน โดยกรอกชื่อฐานข้อมูล และคลิกที่ปุ่ม Create

3. เรายังไม่สร้างตารางในฐานข้อมูล เนื่องจากเราจะทำการนำเข้าฐานข้อมูลจากไฟล์แทน โดยให้เราคลิกที่ลิงก์ Import บนหน้าแดชบอร์ด และคลิกที่ปุ่ม Browse ที่ช่อง File to import เพื่อเลือกไฟล์ฐานข้อมูลที่ต้องการนำเข้า จากนั้นคลิกที่ปุ่ม Go บริเวณด้านล่าง เพื่อเริ่มกระบวนการนำเข้าฐานข้อมูล

4. เพียงเท่านี้เราจะได้ฐานข้อมูลที่ประกอบด้วยตารางและข้อมูลเรียบร้อยทั้งหมด โดยให้สังเกตที่ข้อความ Import has been successfully finished



รูปที่ 2.57 หน้าจอแสดงการเลือกไฟล์ฐานข้อมูลที่จะนำเข้า

2.6.9 การเขียน PHP จัดการ MySQL

เราได้เรียนรู้การใช้ฐานข้อมูล MySQL และการใช้งานโปรแกรม PhpMyAdmin เพื่อจัดการฐานข้อมูล MySQL ไปแล้ว และถึงแม้ว่าโปรแกรม PhpMyAdmin จะสามารถใช้จัดการฐานข้อมูล MySQL ได้ง่าย แต่ในทางปฏิบัติแล้ว เราคงไม่อนุญาตให้ผู้ใช้งานสามารถเข้ามาใช้โปรแกรม PhpMyAdmin เพื่อจัดการฐานข้อมูล MySQL เพราะมิฉะนั้นแล้วผู้ใช้อาจจะทำการเพิ่มลบ หรือแก้ไขข้อมูลต่างๆ ในฐานข้อมูล โดยปกติในการจัดการฐานข้อมูลจะเป็นหน้าที่ของผู้ดูแลระบบ (Administrator) เท่านั้น วิธีแก้ปัญหาดังกล่าวก็คือ เราจะต้องเขียนโปรแกรมภาษา PHP เพื่อให้สามารถติดต่อและเข้าถึงฐานข้อมูล MySQL

2.6.9.1 การเขียน PHP เชื่อมต่อและปิดการเชื่อมต่อ MySQL

เราทราบแล้วว่าก่อนที่จะเข้าใช้งานฐานข้อมูล MySQL ได้ก็จะต้องรู้ชื่อผู้ใช้ (User name) และรหัสผ่าน (Password) เดียวกัน ดังนั้นการที่เราจะเข้าถึงฐานข้อมูล MySQL ได้โดยเบื้องต้นโปรแกรมภาษา PHP ก็จะต้องเรียนรู้คำสั่งที่ใช้เชื่อมต่อ MySQL และหลังจากเข้าถึง MySQL เสร็จแล้วเราจะจะต้องทำการปิดการเชื่อมต่อ MySQL

คำสั่งเชื่อมต่อ MySQL

`mysqli_connect(host name, user name, password)`

hostname เป็นการระบุชื่อโฮสต์ที่ใช้

user name เป็นการระบุชื่อผู้ใช้เพื่อเข้าไปใช้งาน MySQL

password เป็นการระบุรหัสผ่านเพื่อเข้าไปใช้งาน MySQL

ผลลัพธ์ของคำสั่งนี้ ถ้าสามารถเชื่อมต่อ MySQL ได้จะสั่งผลลัพธ์ตัวแปรการเชื่อมต่อ ซึ่งเราจะนำตัวแปรนี้ไปใช้งานในคำสั่งอื่นต่อไป แต่ถ้าไม่สามารถเชื่อมต่อ MySQL ได้จะส่งผลลัพธ์เป็นค่าเท็จ (False) กลับมาให้

คำสั่งปิดการเชื่อมต่อ MySQL

ถ้าเราต้องการจบการใช้งาน MySQL ก็ให้ใช้คำสั่งปิดการเชื่อมต่อ MySQL ดังนี้

`mysqli_close(ตัวแปรการเชื่อมต่อที่ได้จากคำสั่ง mysqli_connect)`

นอกจากนี้ ในที่นี้จะแนะนำคำสั่งเพื่อใช้ในการแจ้งความผิดพลาดที่เกิดขึ้นจากการเชื่อมต่อ MySQL คือ คำสั่ง `mysqli_error()` โดยจะบอกเหตุผลว่ามีความผิดพลาดอะไรเกิดขึ้นจากการเชื่อมต่อ ดังนั้นเราสามารถนำคำสั่ง `mysqli_error()` มาประยุกต์ใช้กับคำสั่ง `mysqli_connect()`

2.6.9.2 การเขียน PHP เพื่อสร้างฐานข้อมูลและตาราง

โดยปกติเราจะสร้างฐานข้อมูลและตารางจากโปรแกรม PhpMyAdmin แต่เราสามารถเขียนโปรแกรมภาษา PHP เพื่อสร้างฐานข้อมูลและตารางได้ เช่นกัน คำสั่งที่สำคัญที่ใช้ในการจัดการคือคำสั่ง `mysqli_query()` โดยคำสั่งนี้จะรับคำสั่ง SQL และนำมาประมวลผลต่อไป รูปแบบในการใช้งานเป็นดังนี้

`mysqli_query(ตัวแปรการเชื่อมต่อ, คำสั่ง SQL)`

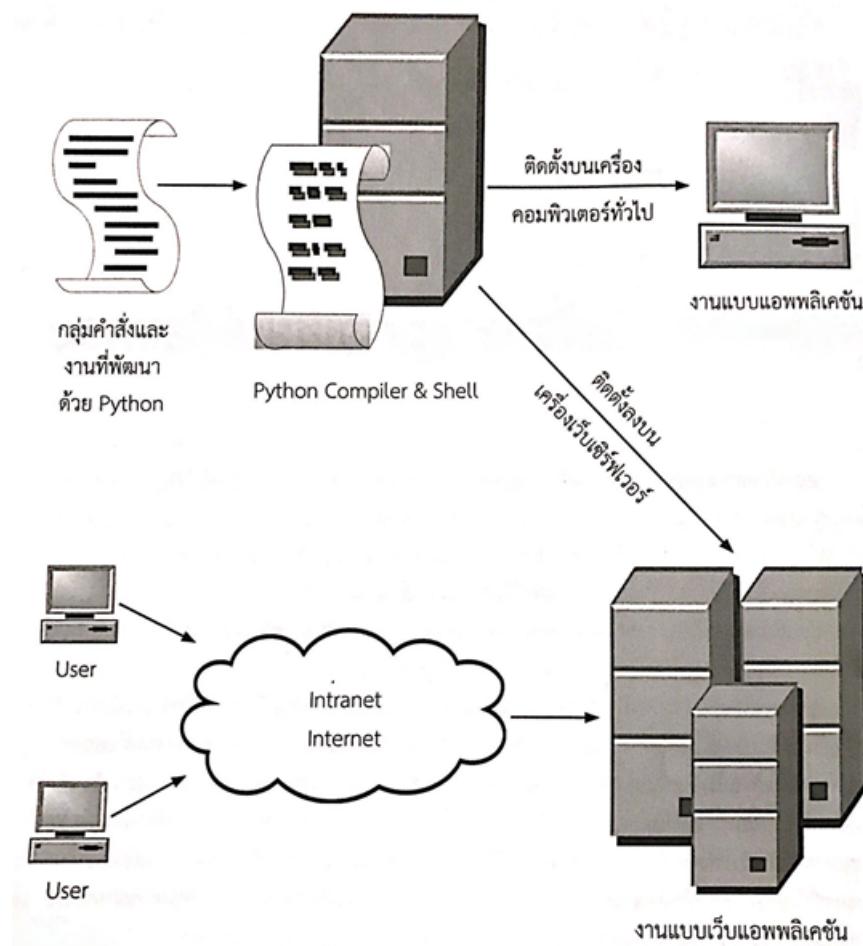
ผลลัพธ์ของคำสั่งนี้จะสั่งค่าจริง (True) เมื่อสามารถทำคำสั่ง SQL ได้ แต่จะส่งค่าเป็นเท็จ (False) เมื่อไม่สามารถทำคำสั่ง SQL ได้ เช่นอาจจะเนื่องมาจาก การพิมพ์คำสั่งที่ผิด เป็นต้น

2.6.9.3 การเขียน PHP เพื่อเข้าถึงข้อมูลในตาราง

ฟังก์ชันที่สำคัญในการเข้าถึงข้อมูลในตาราง คือ การเพิ่มข้อมูล (insert), การลบข้อมูล (delete), การแก้ไขข้อมูล (update) และการแสดงข้อมูลหรือการจัดทำรายงาน (select)

2.7 Python

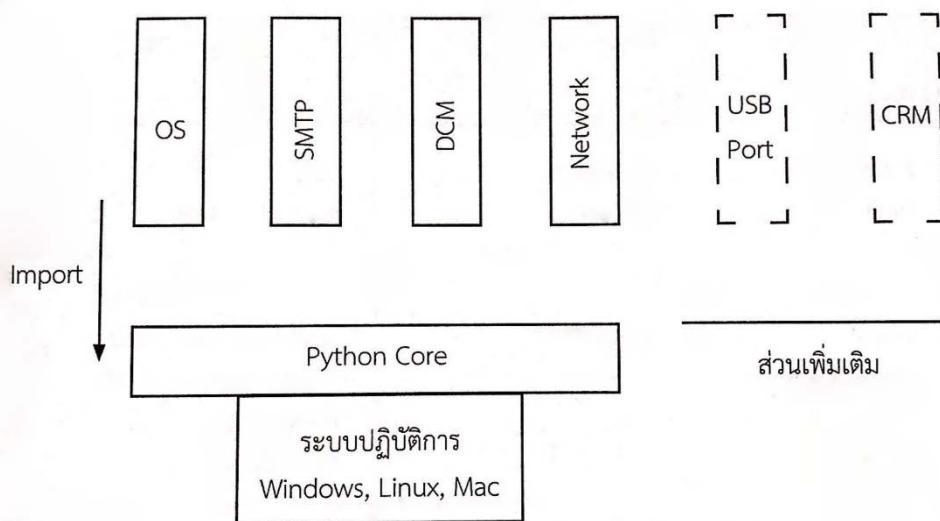
สถาปัตยกรรมของ Python ได้รับการออกแบบให้นำไปพัฒนารองรับงาน แบ่งได้เป็น 2 ประเภท ได้แก่ การพัฒนางานประเภทแอปพลิเคชัน (Application) และเว็บแอปพลิเคชัน (Web Application) ซึ่งวิธีการเขียนโปรแกรม การใช้คำสั่ง รูปแบบการใช้งาน เทคนิคการเขียนทั้งหลาย ไม่ได้แตกต่างกันมาก โดยหากนำภาษา Python ไปสร้างงานเว็บแอปพลิเคชัน ก็ต้องเพิ่มเติมความรู้เกี่ยวกับภาษา HTML, CSS, JavaScript และการจัดการดูแลโปรแกรมบนเว็บเซิร์ฟเวอร์ จึงทำให้การเขียนเว็บโปรแกรมเป็นไปได้อย่างเหมาะสมต่อการนำไปใช้งานในแต่ละ โปรเจกต์



รูปที่ 2.58 แสดงสถาปัตยกรรมความสามารถของภาษา Python

ขอบเขตและขีดความสามารถของ Python ในการนำไปเขียนโปรแกรมใช้งานเชิงลึกร่วมกับไลบรารีมาตรฐาน หรือบางครั้งก็เรียกว่า Standard Module ของ Python (Standard Module เป็นที่อยู่ของกลุ่มคำสั่งที่มีใช้งานใน Python โดยแบ่งตามชนิดของกลุ่มที่จะนำมาใช้งาน เช่น Standard Module เกี่ยวกับ OS ก็จะเป็นกลุ่มคำสั่งในการจัดการเกี่ยวกับ การคัดลอก (Copy), การตัด (Cut)

การสร้างและคำสั่งอื่นๆ สำหรับไฟล์, โฟลเดอร์ในสาร์ดไคร์วของเครื่องคอมพิวเตอร์เรา เป็นต้น) ก็ชื่นอยู่กับงานที่พัฒนาขึ้นมาต้องการให้กับคุณคำสั่งอะไรบ้างสำหรับงานในโปรแกรมที่กำลังพัฒนา ยกตัวอย่าง เราต้องการอ่านข้อมูลจากไฟล์ไคร์ C และทำการจัดส่งอีเมล (E-mail) ให้กับลูกค้า ซึ่ง ในโปรแกรมของ Python จำเป็นต้องเพิ่มโมดูลเกี่ยวกับการจัดการ OS และ โมดูลเกี่ยวกับอีเมล (SMTP) เพื่อเรียกใช้คำสั่งให้ครบถ้วนตามกระบวนการทำงานดังที่กำหนดไว้ในตอนต้น



รูปที่ 2.59 โมดูลหลักที่เป็นมาตรฐานและโมดูลที่ได้เพิ่มเติมให้กับ Python

นอกจากนี้ Python ยังเปิดโอกาสให้เพิ่มเติม โมดูลพิเศษที่นอกเหนือจากที่ Python ได้จัดเตรียมไว้ให้ ซึ่ง โมดูลพิเศษเหล่านี้ก็ได้รับการพัฒนาจาก Python, C, C++, Java, Dot Net หรือภาษาอื่นๆ ด้วยเหตุผลบางประการที่จำเป็นต้องใช้ภาษาอื่นๆ มาช่วยเสริม ร่วมกับ Python ยกตัวอย่าง ถ้าต้องการต่อ Python ร่วมกับ MySQL ซึ่งคำสั่งเชื่อมโยงฐานข้อมูลพิเศษสำหรับ MySQL ไม่ได้ถูกสร้างไว้ให้เรียกใช้ในโมดูลมาตรฐานของ Python เราอาจจะเป็นต้องไปดาวน์โหลด PBM Library พิเศษเฉพาะของ MySQL ที่เว็บไซต์ MySQL (หากฐานข้อมูลสนับสนุนการเชื่อมโยงกับ Python ก็จะมีไลบรารีสำหรับเชื่อมโยงฐานข้อมูลให้ดาวน์โหลดมาติดตั้งและใช้งานเพิ่มเติม)

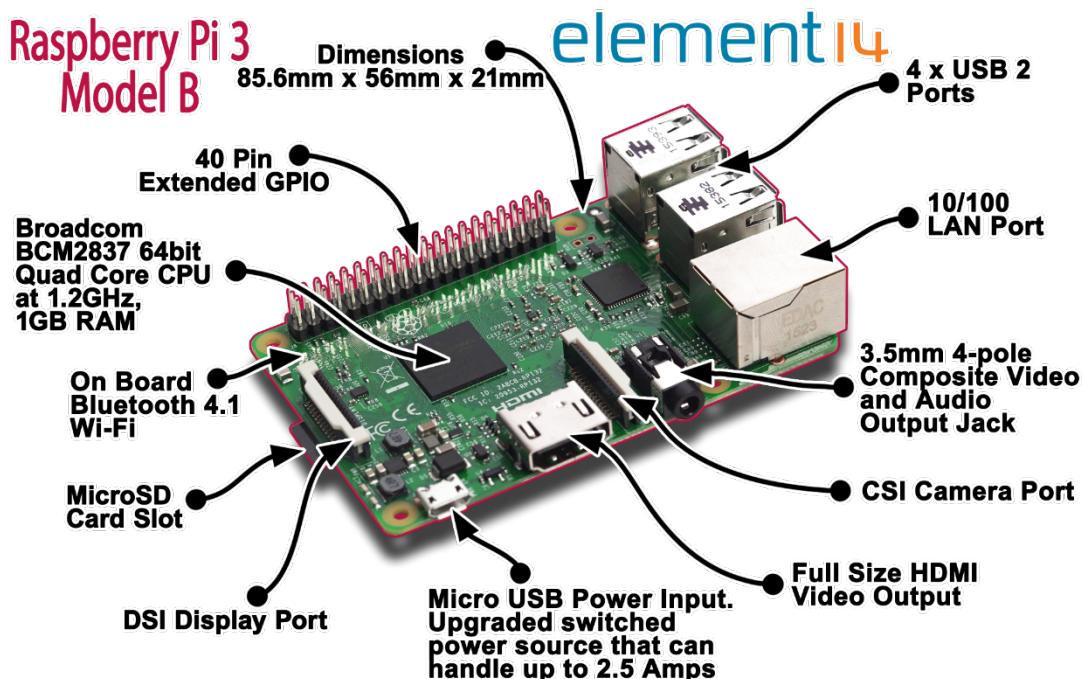
ตารางที่ 2.6 แสดงรายชื่อโมดูลมาตรฐานของ Python

กลุ่ม	ชื่อโมดูลมาตรฐานของ Python	ประเภทการใช้งาน
ตัวเลข (Number)	datetime decimal	วันที่ ตัวเลขแบบบีดพิเศษ
การคำนวณ (Math)	math cmath	การคำนวณทางคณิตศาสตร์ การคำนวณจำนวนเชิงซ้อน
อาร์กิวเมนต์ (Argument)	sys	การผ่านค่าอาร์กิวเมนต์ของแต่ละโมดูล
เวลา (Time)	time	กลุ่มคำสั่งเกี่ยวกับเวลา
การสุ่มค่า	random	กลุ่มคำสั่งที่สุ่มค่าตัวเลข
Local Path / Directory	OS	กลุ่มคำสั่งเกี่ยวกับ Path และ Directory ของระบบปฏิบัติการ
ข้อมูลรูปแบบ XML	xml	กลุ่มคำสั่งจัดการคุณลักษณะข้อมูลแบบ XML
อีเมล	smtplib, mimetypes, e-mail, imaplib	กลุ่มคำสั่งสำหรับจัดการ จัดการอีเมล
เครือข่ายการสื่อสาร	socket, socketserver	กลุ่มคำสั่งจัดการ โปรโตคอลและการสื่อสาร

ทั้งนี้ภาษา Python จัดเป็นภาษาที่อยู่ในระดับสูงเทียบกับภาษา Visual Basic บางกรณีที่ต้องการใช้งานเชิงลึกในระดับ Kernal ของระบบปฏิบัติการ การเรียกใช้พอร์ตฮาร์ดแวร์ต่างๆ การจัดการระดับหน่วยความจำ การเขียนโปรแกรมระบบเครือข่ายเชิงลึก ตามที่กล่าวไว้บางส่วน ภาษา Python ไม่มีความสามารถเข้าไปจัดการได้เลย ต้องอาศัยภาษาอื่นๆ ที่มีความสามารถในการเขียนโปรแกรมเชิงลึก เช่น C, C++, Java ฯลฯ สร้างโมดูลพิเศษไว้ชื่อมต่อ กับ Python ที่จะเรียกใช้ในลักษณะเป็นฟังก์ชันย่อยภายในของโมดูลพิเศษเหล่านั้นอีกที

2.8 Raspberry Pi

Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็ก ออกแบบและพัฒนาโดย Raspberry Pi Foundation ที่รองรับระบบปฏิบัติการ Linux Raspbian ตัวบอร์ดมี Wi-Fi, Bluetooth, USB, Ethernet Port, ช่องเสียบต่อจอภาพแบบ Micro HDMI มีช่องเสียบหูฟัง และขาอินพุต เอาต์พุต ช่อง USB สำหรับเสียบแป้นพิมพ์ เม้าส์ หรืออุปกรณ์อื่น ๆ เช่น กล้อง ระบบ GPS หรือ เซนเซอร์ต่าง ๆ ฯลฯ โดยคุณสมบัติของ Raspberry pi 3 Model B สามารถแสดงได้ดังตารางที่ 3.1



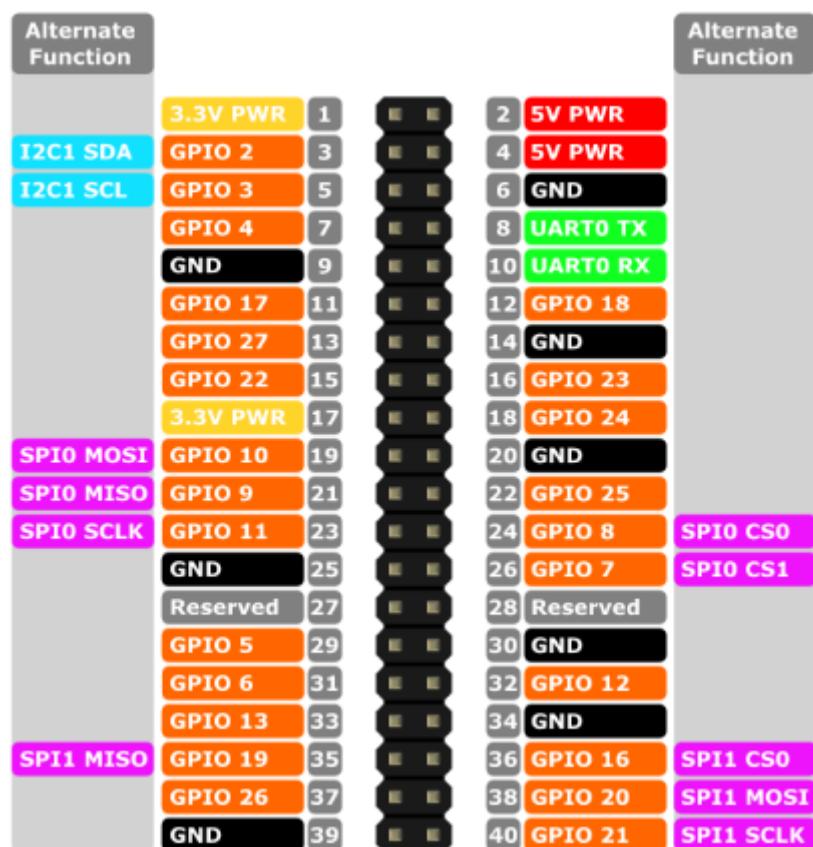
รูปที่ 2.60 ตำแหน่งของอุปกรณ์ต่าง ๆ บนบอร์ด Pi3

ตารางที่ 2.7 คุณสมบัติของ Raspberry pi 3 Model B

Size	86mm x 56mm x 21mm
Processor	Broadcom BCM2837 SoC 1.2 GHz 64-bit quad-core ARM Cortex-A53
Memory	1 GB LPDDR2
Storage	MicroSD slot
WLAN	802.11n
Ethernet	10/100 Mbit/s
Bluetooth	Bluetooth 4.1, Bluetooth Low Energy (BLE)

GPU	Broadcom VideoCore IV
Video	HDMI, Composite
Audio	via HDMI via 3.5 mm phone jack
USB	USB 2.0 4 ports
GPIO	40 pins
Power Source	5V via MicroUSB or GPIO header
Power Rating	800 mA (4.0 W)

จุดเชื่อมต่อ GPIO(General Purpose Input/Output) ของ Raspberry pi สามารถกำหนดให้เป็น Input หรือ Output ได้ ซึ่งทำให้ควบคุมได้แต่ละ pin เมื่อ่อนไม่โครคณ์ โตรลเลอร์ GPIO pin ทั้งหมดมี 26 pin โดยใน pin นั้นจะมีฟังก์ชันต่าง ๆ ตามมาด้วย ได้แก่ SPI, I2C, serial และยังมี pin ที่สามารถเชื่อมต่อ UART, 3V3 , 5V power และ GND เป็น 40 pin ซึ่งแสดงดังรูปที่ 2.59



รูปที่ 2.61 ตำแหน่งขา GPIO 40 PIN

2.8.1 DHT22 Temperature and Humidity Sensor

DHT22 เป็นโมดูลวัดอุณหภูมิ (Temperature) และความชื้นสัมพัทธ์ (Humidity) ที่มีความแม่นยำสูงในการวัด โดยใช้ชิปDHT22 ให้ Output ออกมาเป็นแบบ Digital Output โ้มดูลมาพร้อม PCB และสายไฟ ต่อ R4.7K เรียบร้อยต่อสายใช้งานได้ทันที โดยมีรายละเอียดดังนี้

- ขนาด 28มม. X 12มม. X 10มม.
- รองรับแรงดันไฟฟ้าตั้งแต่ 3.3V - 6V
- วัดอุณหภูมิได้ตั้งแต่ -40 ถึง 80 องศาเซลเซียส
- วัดค่าความชื้นได้ตั้งแต่ 0 - 100%
- ค่าความผิดพลาดในการวัดอุณหภูมิเพียง $+0.5$ องศาเซลเซียส
- ค่าความผิดพลาดในการวัดความชื้นเพียง 2%



รูปที่ 2.62 DHT22 Temperature and Humidity Sensor

2.8.1.1 การใช้งาน DHT22 ร่วมกับ Raspberry Pi เป็นต้น

- 1) ทำการต่อวงจรดังรูปที่ 2.61 เพื่อเชื่อมต่อระหว่าง Raspberry pi กับ DHT22
- 2) เปิดหน้า terminal ของ raspberry pi เพื่อทำการติดตั้ง Library ของ DHT22 ด้วยคำสั่งต่อไปนี้

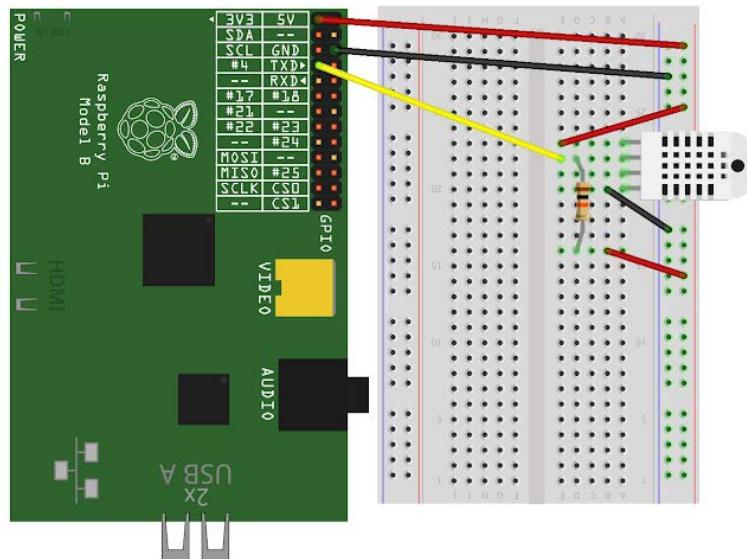
- sudo apt-get update
- sudo apt-get install build-essential python-dev
- git clone https://github.com/adafruit/Adafruit_Python_DHT.git
- cd Adafruit_Python_DHT
- sudo python setup.py install

- 3)ทดสอบอ่านค่าจาก DHT 22 ด้วยคำสั่งต่อไปนี้

- cd example

- sudo ./AdafruitDHT.py 22 4

4) จะได้ค่าอุณหภูมิ และ ความชื้นในอากาศดังรูปที่ 2.62



รูปที่ 2.63 การต่อเข็มข่าย DHT22 กับ Raspberry pi

```
pi@raspberrypi:~/Adafruit_Python_DHT/examples
File Edit Tabs Help
pi@raspberrypi:~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 22 4
Temp=32.0* Humidity=60.5%
pi@raspberrypi:~/Adafruit_Python_DHT/examples $
```

รูปที่ 2.64 ค่าอุณหภูมิและความชื้นในอากาศที่วัดได้

2.8.2 Ambient Light Sensor Module GY-302 (BH1750FVI)

เซนเซอร์วัดความเข้มแสง BH1750FVI ให้ค่าเอาต์พุตเป็นความเข้มแสงหน่วยเป็น lux สามารถวัดค่าความเข้มแสงได้ในช่วง 0- 65535 lux ทำงานโดยใช้การสื่อสารแบบ I²C ใช้แรงดันไฟฟ้า 3 - 5V มีขนาดเพียง 13.9 มม. X 18.5 มม.



รูปที่ 2.65 Ambient Light Sensor Module GY-302

2.8.2.1 การใช้งาน BH1750FVI ร่วมกับ Raspberry Pi เปื้องต้น

เนื่องจาก Raspberry pi มีคุณสมบัติที่สามารถเชื่อมต่ออุปกรณ์ผ่าน Bus I²C ได้ แต่ต้องทำการติดตั้ง Library ของ Bus I²C ก่อน โดยมีขั้นตอนดังต่อไปนี้

- 1) ทำการต่อวงจรดังรูปที่ 2.64 เพื่อเชื่อมต่อระหว่าง Raspberry pi กับ BH1750FV
- 2) เปิดหน้า terminal ของ raspberry pi
 - \$ sudo apt-get install python3-smbus
 - \$ sudo apt-get install i2c-tools
 - \$ sudo reboot
- 3) เมื่อ Reboot เครื่องแล้ว ให้ Enable Bus i2c ของ Raspberry Pi ด้วยคำสั่ง
 - \$ sudo raspi-config
 - จะปรากฏหน้าต่าง Raspberry Pi Software Configuration Tools ขึ้นมา จากนั้นเลือก Advance Option >> I2C >> Yes >> OK
- 4) จากนั้น เขียน Script Python เพื่อวัดค่าความเข้มแสง ดังต่อไปนี้

```
import time
import smbus
import datetime
```

```

bus = smbus.SMBus(1) #(512MB)

addr = 0x23 # i2c adress

while True:

    data = bus.read_i2c_block_data(addr,0x11)

    lum=(data[1] + (data[0]<<8) / 1.2)

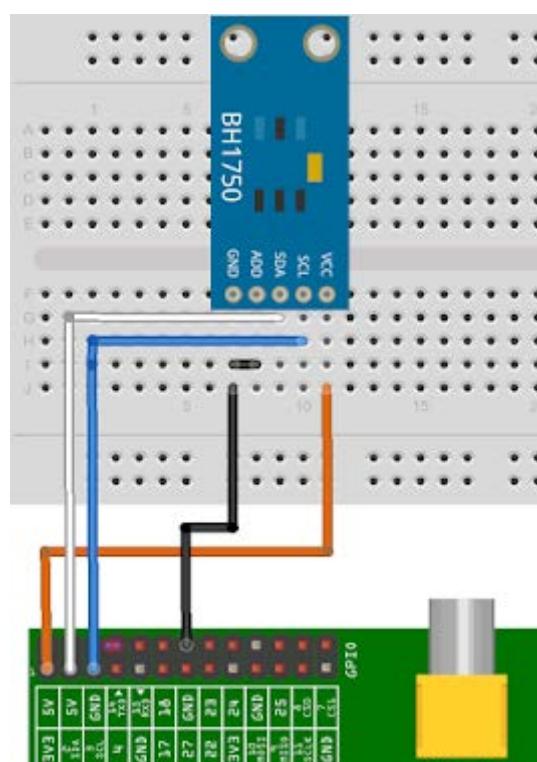
    date=str(datetime.datetime.now())

    print ("Lum = ",lum)

    time.sleep(0.5)

```

5) Run Program เพื่อทดสอบ จะได้ผลลัพธ์ตามรูปที่ 2.65



รูปที่ 2.66 การเชื่อมต่อ BH1750FV กับ Raspberry pi

```

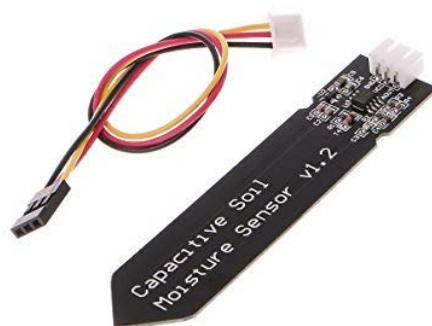
pi@raspberrypi:~/Desktop $ python BHtest.py
('Lum = ', 46.0)
('Lum = ', 46.0)
('Lum = ', 39.0)
('Lum = ', 29.0)
('Lum = ', 27.0)
('Lum = ', 27.0)
('Lum = ', 26.0)
('Lum = ', 28.0)
('Lum = ', 28.0)
('Lum = ', 28.0)
('Lum = ', 29.0)
('Lum = ', 14.0)
('Lum = ', 15.0)
('Lum = ', 24.0)
('Lum = ', 16.0)
('Lum = ', 25.0)
('Lum = ', 31.0)
('Lum = ', 23.0)
('Lum = ', 18.0)
('Lum = ', 19.0)
('Lum = ', 39.0)

```

รูปที่ 2.67 ค่าความเข้มแสงที่วัดได้

2.8.3 Capacitive Analog Soil Moisture Sensor

เซ็นเซอร์วัดความชื้นในดินแบบ Capacitive Analog Soil Moisture Sensor นี้อาศัยหลักการแบบเดียวกับตัวเก็บประจุ คือเมื่อเซ็นเซอร์ได้รับความชื้นจะทำหน้าที่เป็นตัวเก็บประจุไฟฟ้า ยิ่งมีความชื้นมาก จะยิ่งทำให้ค่าประจุไฟฟ้ามากขึ้นเนื่องจากระยะห่างระหว่างขั้วทั้ง 2 แนบสนิทมากขึ้น โดยค่าเอาต์พุตที่ได้จะเป็นแรงดันไฟฟ้า หากใช้ Raspberry Pi จำเป็นต้องต่อ IC MCP3208 เพิ่มเพื่อให้ Raspberry Pi สามารถอ่านค่า Analog ได้ แรงดันเอาต์พุต (Vout) 0-3V แรงดันอินพุต(Vin) 3.3-5.5V



รูปที่ 2.68 Capacitive Analog Soil Moisture Sensor

2.8.4 จอ LCD 16x4 Character

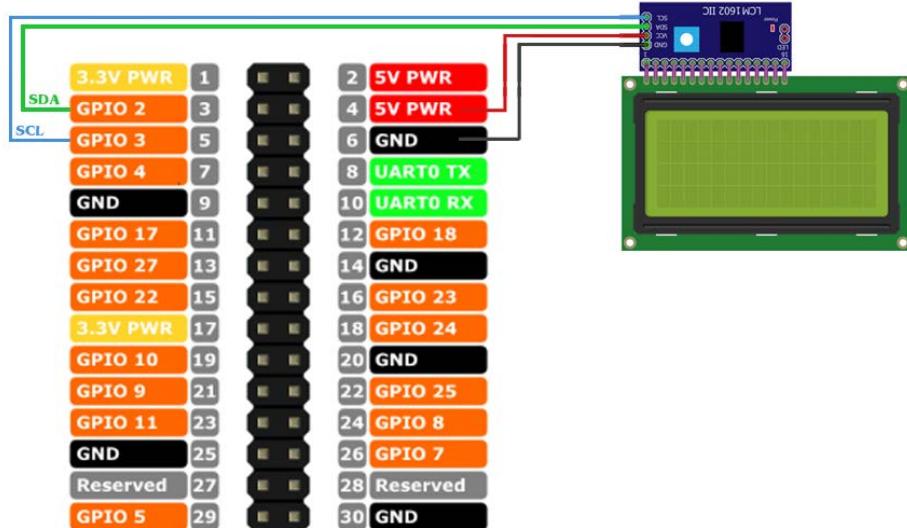
เป็นจอแสดงผลขนาด 16x4 ตัวอักษร ที่แสดงผลเป็นตัวอักษรตามช่องแบบตัวต่อตัว โครงสร้างของLCDจะประกอบด้วยแผ่นแก้ว 2 แผ่นประกบกันเว้นช่องว่างตรงกลางไว้ประมาณ 6-10 ไมโครเมตรด้านในจะเคลือบด้วยตัวนำไฟฟ้าแบบใส่ในระหว่างตัวนำไฟฟ้าแบบใส่จะมีโโมเลกุล พลิกรวมตัวกันในทิศทางที่แสงส่องผ่าน โดยจะใช้การเชื่อมต่ออนุกรม (Serial) แบบ I2Cเพื่อ ติดต่อสื่อสาร



รูปที่ 2.69 จอ LCD แบบ16*4

2.8.4.1 การใช้งาน LCD 16x4 Character ผ่าน I2C ร่วมกับ Raspberry Pi เป็นง่ายด้วย

- 1) ทำการต่อวงจรดังรูปที่ x.x เพื่อเชื่อมต่อระหว่าง Raspberry pi กับ LCD 16*4



รูปที่ 2.70 การเชื่อมต่อ BH1750FV กับ Raspberry pi

2) ทำการดาวน์โหลด Library จาก

<https://gist.github.com/DenisFromHR/cc863375a6e19dce359d>

3) ทำการตรวจสอบ Address ของ Bus I²C หากไม่ถูกต้องให้ทำการเปลี่ยน Address ตามในรูปที่ 2.69

```
"""
# i2c bus (0 -- original Pi, 1 -- Rev 2 Pi)
I2CBUS = 1

# LCD Address
ADDRESS = 0x27

import smbus
from time import sleep
class i2c_device:
```

รูปที่ 2.71 Address I²C ของ LCD

4) จากนั้น เขียน Script Python เพื่อแสดงผลบนจอ LCD ดังตัวอย่าง

```
import RPi_I2C_Driver
from time import *
mylcd = RPi_I2C_Driver.lcd()
mylcd.lcd_display_string("Hello world !", 1)
```

5) Run Program เพื่อทดสอบ จะได้ผลลัพธ์ตามรูปที่ 2.70



รูปที่ 2.72 แสดงข้อความบนจอ LCD

2.8.5 LED Grow Light

หลอดไฟ LED Light Grow คือหลอดที่ผลิตขึ้นมาเพื่อใช้สำหรับปลูกต้นไม้โดยเฉพาะ ใช้สำหรับเร่งการเจริญเติมโตของต้นอ่อนป่องกันลำต้นยืด ใช้แทนแสงอาทิตย์สำหรับปลูกต้นไม้ภายในบ้าน โดยหลอดไฟ LED Light Grow นี้จะใช้เทคโนโลยีมาใช้ในการผลิต โดยผลิต LED ขึ้นมาโดยผลิตแสงความยาวคลื่นในเฉพาะแสงย่างที่พืชมีความต้องการในการเจริญเติบโต ซึ่งคุณสมบัติพิเศษของตัวหลอด LED จะขับเฉพาะความยาวคลื่นแสงที่พืชต้องการได้ซึ่งในกรณีนี้เราจะเลือกความยาวของคลื่นแสงที่ประมาณ 430-460nm และ 630-850nm เนื่องจากว่า ความยาวคลื่นแสงในช่วงนี้เหมาะสมสำหรับการสังเคราะห์แสงของต้นไม้มากที่สุด และช่วยในการเจริญเติบโตของต้นไม้มากที่สุด

ประโยชน์ของแสงสีน้ำเงิน (ความยาวคลื่นแสงประมาณ 430-460nm)

- เป็นช่วงความยาวคลื่นแสงที่สามารถดูมีสีได้มากที่สุด
- กระตุ้นการผลิต Chlorophyll ทำให้ต้นไม้สามารถสังเคราะห์แสงได้มากยิ่งขึ้น
- เร่งการเจริญเติบโตของลำต้น ช่วยให้ลำต้นแข็งแรง และช่วยลดปัญหาลำต้นยึดผิดรูป
- ช่วยให้ใบไม้แข็งแรงและมีสีเขียวสด สวายงาม

ประโยชน์ของแสงสีแดง (ความยาวคลื่นแสงประมาณ 630-660nm)

- เป็นช่วงความยาวคลื่นแสงที่สามารถดูมีสีได้
- เร่งดอก เร่งผล ช่วยบำรุงดอกและผลให้สมบูรณ์ และช่วยขยายขนาดของผลผลิต
- เร่งการเจริญเติบโตของราก และช่วยให้รากแข็งแรง
- เร่งการเจริญเติบโตของลำต้น

โดยในโครงการจะเลือกใช้ LED Grow Light ขนาด 10 W 12V ซึ่งมีรายละเอียดตามตารางที่ 2.8



รูปที่ 2.73 ลักษณะของ LED Grow Light ขนาด 10 W 12V

ตารางที่ 2.8 รายละเอียดของ LED Grow Light ขนาด 10 W 12V

Power	10W
Voltage	9-12V DC
Current	1000mA
Color	full spectrum
Wavelength	400nm~850nm
Luminous Flux	200-300LM
Chip Brand	Epistar
Lifespan	50000 hours lifespan
lighting angle	120~140 degree

2.8.6 Switching Power Supply

Switching Power Supply เป็นแหล่งจ่ายไฟตรงคงค่าแรงดันแบบหนึ่ง และสามารถเปลี่ยนแรงดันไฟจากไปกลับไว้ได้สูง ให้เป็นแรงดันไฟตรงค่าต่ำ เพื่อใช้ในงานอิเล็กทรอนิกส์ได้ เช่น เดิมกันแหล่งจ่ายไฟเชิงเส้น (Linear Power Supply) ถึงแม้พาวเวอร์ซัพพลายทั้งสองแบบจะต้องมี

การใช้หม้อแปลงในการลดทอนแรงดันสูงให้เป็นแรงดันต่ำ เช่นเดียวกัน แต่สวิตซ์เพาเวอร์ซัพพลายจะใช้หม้อแปลงที่มีขนาดเล็ก และน้ำหนักน้อย เมื่อเทียบกับแหล่งจ่ายไฟเชิงเส้น อีกทั้งสวิตซ์เพาเวอร์ซัพพลายยังมีประสิทธิภาพสูงกว่าอีกด้วย ในโครงงานนี้จะเลือกใช้ Switching Power Supply ขนาด 12V 40A ซึ่งมีรายละเอียดตามตารางที่ 2.9



รูปที่ 2.74 ลักษณะของ Switching Power Supply

ตารางที่ 2.9 รายละเอียดของ Switching Power Supply ขนาด 12V 40A

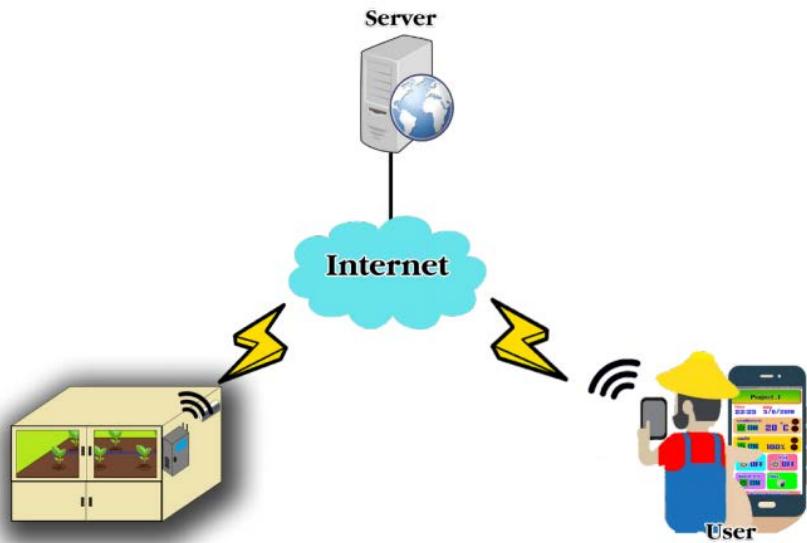
Input	110-220V AC
Output Voltage	12V DC
Output power	480W
Output Current	40A
Dimension	113*215*50 mm.
Material	Aluminum
Current ripple	>120mV
Other qualifications	Overload protection, over-current protection, resistance to impact and start electronic jamming
Weight	1000 g.

บทที่ 3

การสร้างและการออกแบบระบบ

3.1 ภาพรวมการทำงานของระบบ

ภาพรวมของระบบจะประกอบไปด้วย ตู้จำลองโรงเรือนแบบปิด ซึ่งจะใช้สำหรับเพาะปลูก และติดตั้งอุปกรณ์เซ็นเซอร์ที่จะใช้วัดต่าง ๆ เช่นอุณหภูมิ กับ Raspberry pi เพื่อใช้ควบคุมและส่งข้อมูลไปยังระบบอินเตอร์เน็ต จากนั้นจะถูกจัดเก็บไว้ใน Server เมื่อผู้ใช้ต้องการควบคุมหรือดูข้อมูลสามารถดูข้อมูลผ่าน Web Application จากคอมพิวเตอร์หรือสมาร์ทโฟน



ภาพที่ 3.1 ภาพรวมของระบบควบคุมโรงเรือนสมาร์ทฟาร์มแบบปิด

สำหรับผลลัพธ์ที่ออกมานี้จะเป็นรูปแบบของ Web Browser ซึ่งสามารถใช้ผ่านระบบมือถือเพื่อเรียกคุ้มข้อมูลการทำงานและสามารถตั้งค่าการทำงานของฟาร์มอัจฉริยะ ได้รวมถึงสามารถเรียกดูรายละเอียดในการทำงานของฟาร์มอัจฉริยะเพื่อเป็นข้อมูลในการปรับปรุงค่าการทำงานของฟาร์มอัจฉริยะให้เหมาะสมกับพืชที่ปลูกได้อย่างเหมาะสม โดยในที่นี้จะแบ่งเป็น 2 ส่วนดังนี้

- 1) ส่วนระบบควบคุมโรงเรือนสมาร์ทฟาร์มแบบปิด
- 2) ส่วนฐานข้อมูล และ เว็บแอปพลิเคชัน

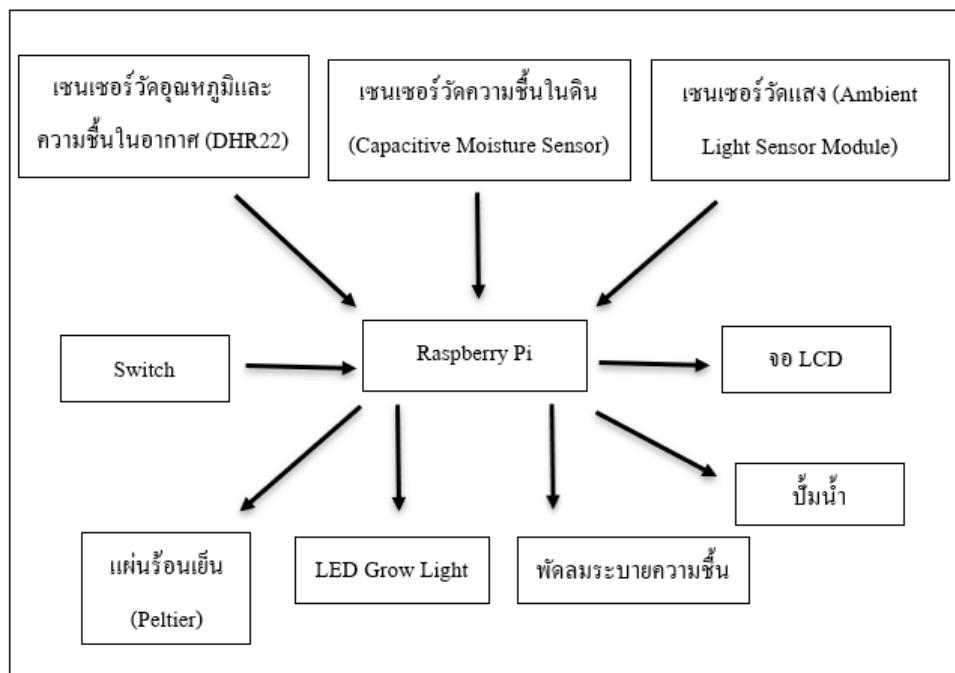
โดยการแยกแต่ละส่วนนั้นสามารถดูได้จากภาพการทำงานของระบบควบคุมโรงเรือนสมาร์ทฟาร์มแบบปิดจะถูกติดตั้งโดยมีส่วนของตู้ control เป็นตัวควบคุมโดยตรงและจะส่งข้อมูลการตั้งค่าสถานะต่างๆ ผ่านระบบ Web Application หลังจากนั้นจะมีการส่งข้อมูลจาก Web

application ไปยังตัวรับสัญญาณอินเทอร์เน็ตเพื่อรับค่าข้อมูลจาก Web Application คือ raspberry pi ซึ่ง raspberry pi จะเป็นตัวส่งข้อมูลไปยังอุปกรณ์ต่างๆเพื่อส่งงานให้อุปกรณ์ทำงานตามที่ผู้ใช้ได้ตั้งค่าผ่าน web Application นั้นๆได้

3.2 การออกแบบตู้จำลองโรงเรือนแบบปิด

ผู้ใช้งานสามารถควบคุมการทำงานได้จากปุ่มกดที่อยู่หน้าตู้คอนโทรล ซึ่งจะสามารถควบคุมระบบได้ดังนี้

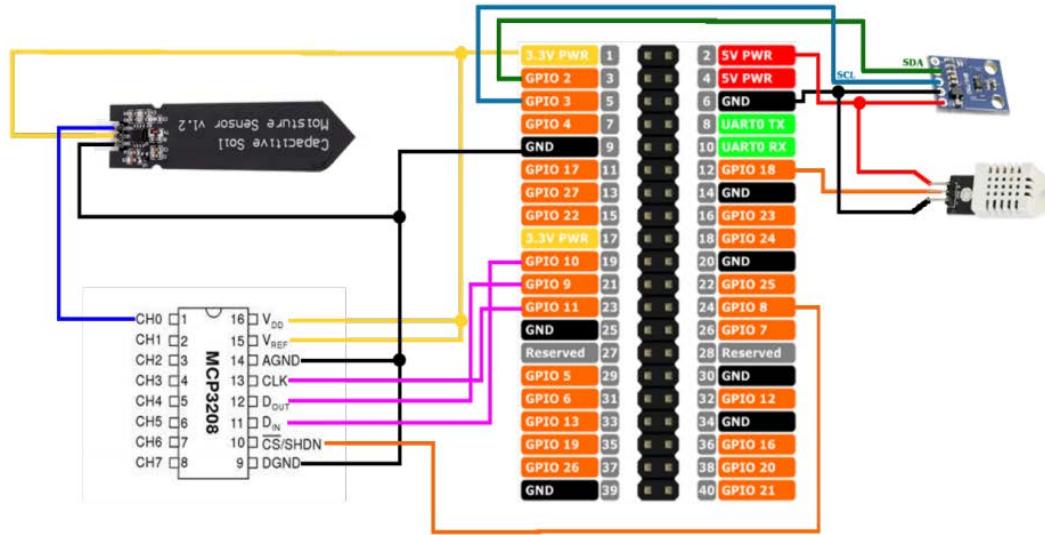
- 1) สามารถควบคุมการเปิด - ปิดไฟได้
- 2) สามารถควบคุมเพิ่ม - ลดอุณหภูมิได้
- 3) สามารถเปิด - ปิดระบบบรอน้ำได้
- 4) สามารถเปิด - ปิดพัดลมระบายความชื้น
- 5) สามารถดูค่าอุณหภูมิ , ความชื้นในอากาศ , ความชื้นในดิน และความเข้มแสง ได้จาก LCD หน้าตู้คอนโทรล



รูปที่ 3.2 ภาพแสดงการส่วนการทำงานของระบบโรงเรือนสมาร์ทฟาร์มแบบปิด

การออกแบบวงจรออกแบบเป็นวงจรโดยการเชื่อมต่อกับโมดูลแต่ละโมดูลของแต่ละอุปกรณ์ โดยใช้ raspberry pi เป็นอุปกรณ์เชื่อมต่ออินเทอร์เน็ตเพื่อใช้ในการรับข้อมูลจาก Web Application ในการกำหนดค่าสถานะและการทำงานต่างๆของระบบควบคุมโรงเรือน Smart Farm แบบปิดได้

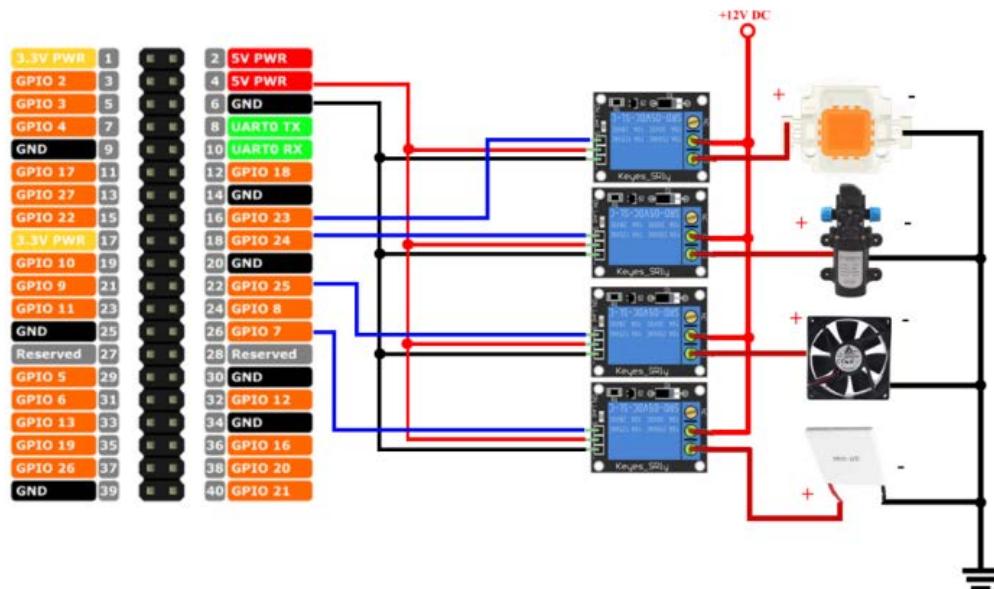
3.2.1 วงจรอ่านค่าจากเซ็นเซอร์ต่างๆ



รูปที่ 3.3 การต่อวงจรอ่านค่าเซ็นเซอร์

วงจรนี้เป็นการต่ออุปกรณ์เซ็นเซอร์ที่ใช้ในการอ่านค่าโดย โมดูล DHT 22 สามารถต่อเข้ากับ GPIO1 8 (ขา12) ได้เลย เพราะให้อาต์พุตเป็นดิจิตอล ส่วนโมดูล Capacitive Analog Soil Moisture Sensor ให้อาต์พุตเป็นอนาล็อก จึงต้องต่อร่วมกับ ไอซี MCP3208 เพื่อใช้แปลงสัญญาณอนาล็อกเป็นดิจิตอล เพราะ Raspberry Pi ไม่สามารถอ่านค่าอนาล็อกได้โดยตรง ส่วนเซ็นเซอร์วัดความชื้นแสง B H1 750 F VI ใช้การสื่อสารแบบ I2C สามารถต่อเข้าได้เลยโดย SDA ต่อเข้ากับ GP IO2(ขา3) S C L ต่อเข้ากับ GP IO4(ขา15)

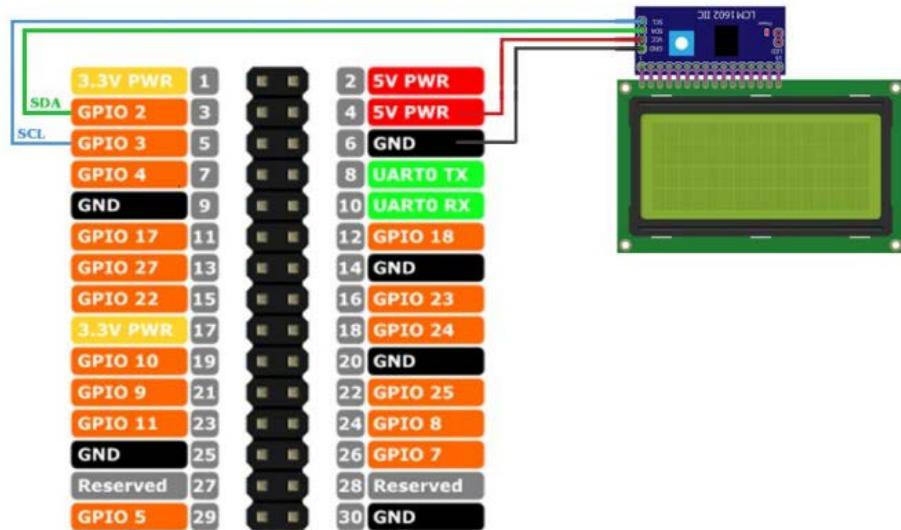
3.2.2 วงจรควบคุมอ่าต์พุต



รูปที่ 3.4 การต่อวงจรควบคุมอ่าต์พุต

วงจรอารควบคุมปั๊มน้ำ พัดลมดูดอากาศ แฟ่นเพลเทียร และไฟ LED Grow Light ต้องอาศัยการสั่งงานผ่านรีเลย์ เพราะอุปกรณ์ใช้แรงดันไฟ 12 โวลต์และกินกระแสมาก แรงดันไฟและกระแสไฟจากขา GP IO ของ Raspberry Pi ไม่เพียงพอต่อการทำการของอุปกรณ์ จึงต้องนำรีเลย์มาต่อเพื่อควบคุมการทำงานของอุปกรณ์ทั้งหมด

3.2.3 การต่อจอ LCD 16*4

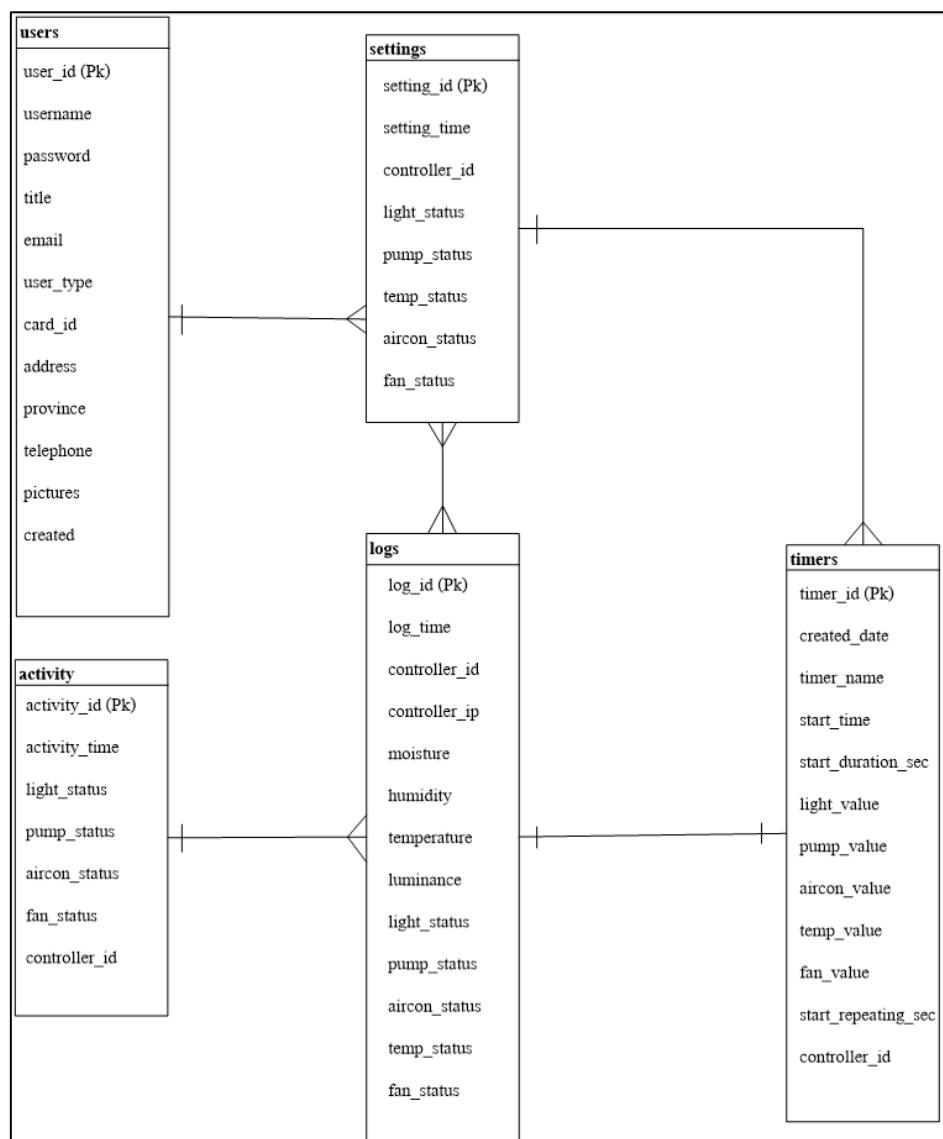


รูปที่ 3.5 การต่อจอ LCD เข้ากับ Raspberry pi

การต่อจอ LCD เข้ากับ Raspberry pi โดยใช้ I2 C สามารถต่อเข้าได้โดย SDA ต่อเข้ากับ GP IO2(ขา3) S CL ต่อเข้ากับ GP IO4(ขา5) แต่หากมีการใช้ I2 C ซึ่งกันต้องทำการตรวจสอบ Address ของอุปกรณ์นั้น ๆ หากมีการทำงานซ้ำจะทำการสั่งให้ทำงานเปลี่ยนAddress ก่อน

3.3 ส่วนฐานข้อมูล และ เว็บแอปพลิเคชัน

ส่วนของเว็บแอปพลิเคชันจะเป็นส่วนที่ใช้ในการตั้งค่าระบบการทำงานของระบบควบคุม โรงเรือน Smart Farm แบบปิดให้สามารถตั้งค่าได้ตั้งแต่การควบคุมอุณหภูมิ ความชื้นในดิน และ ความชื้นในอากาศเพื่อให้สามารถควบคุมการทำงานให้เหมาะสมกับพืชแต่ละชนิดที่ปลูกโดย สามารถตั้งค่าสถานะต่างๆ ผ่านเว็บไซต์โดยเว็บไซต์จะทำหน้าที่เก็บข้อมูลการทำงานทั้งหมดและ สามารถอกรายงานการทำงานของระบบควบคุม โรงเรือน Smart Farm แบบปิดได้โดยผู้ใช้ สามารถเลือกดูข้อมูลการทำงานย้อนหลังผ่านระบบเว็บไซต์และทำการบันทึกข้อมูลต่างๆ ลงใน ฐานข้อมูลได้



รูปที่ 3.6 data base Schema

ในส่วนของการออกแบบฐานข้อมูล ประกอบด้วย 5 Entity ดังนี้

3.3.1 Entity User ประกอบด้วย Attribute

- user_id (Primary) หมายเลขผู้ใช้ เป็น Primary key
- username ชื่อผู้ใช้งาน
- password รหัสของผู้ใช้งาน
- title ชื่อตำแหน่งผู้ใช้งาน
- email อีเมลผู้ของใช้งาน
- user_type สถานะของผู้ใช้งาน
- card_id รหัสประชาชน
- address บ้านเลขที่
- province จังหวัด
- telephone เบอร์โทรศัพท์
- pictures รูป
- created

3.3.2 Entity settings ประกอบด้วย Attribute

- setting_id (Pk) หมายเลขการตั้งค่า เป็น Primary key
- setting_time เวลาที่ตั้ง
- controller_id เลขอุปกรณ์
- light_status ค่าสถานะ เปิด/ปิด ไฟ
- pump_status ค่าสถานะ เปิด/ปิด ปั๊ม
- temp_status ค่าอุณหภูมิ
- aircon_status ค่าสถานะ เปิด/ปิด ทำความเย็น
- fan_status ค่าสถานะ เปิด/ปิด พัดลมระบบ

3.3.3 Entity activity บันทึกตอนที่ผู้ใช้งานตั้งค่า ประกอบด้วย Attribute

- activity_id (Pk) หมายเลขกิจกรรม เป็น Primary key
- activity_time เวลา
- light_status ค่าสถานะ เปิด/ปิด ไฟ
- pump_status ค่าสถานะ เปิด/ปิด ปั๊ม

- aircon_status ค่าสถานะ เปิด/ปิด ทำความเย็น
- fan_status ค่าสถานะ เปิด/ปิด พัดลมระบบ
- controller_id เลขอุปกรณ์

3.3.4 Entity logs บันทึกการทำงาน (log) ประกอบด้วย Attribute

- log_id (Pk) หมายเลข log เป็น Primary key
- log_time เวลา
- controller_id เลขอุปกรณ์
- controller_ip ip อุปกรณ์
- moisture ความชื้นในดิน
- humidity ความชื้นาากาศ
- temperature อุณหภูมิ
- luminance ความสว่าง
- light_status ค่าสถานะ เปิด/ปิด ไฟ
- pump_status ค่าสถานะ เปิด/ปิด ปั๊ม
- aircon_status ค่าสถานะ เปิด/ปิด ทำความเย็น
- temp_status ค่าอุณหภูมิ
- fan_status ค่าสถานะ เปิด/ปิด พัดลมระบบ

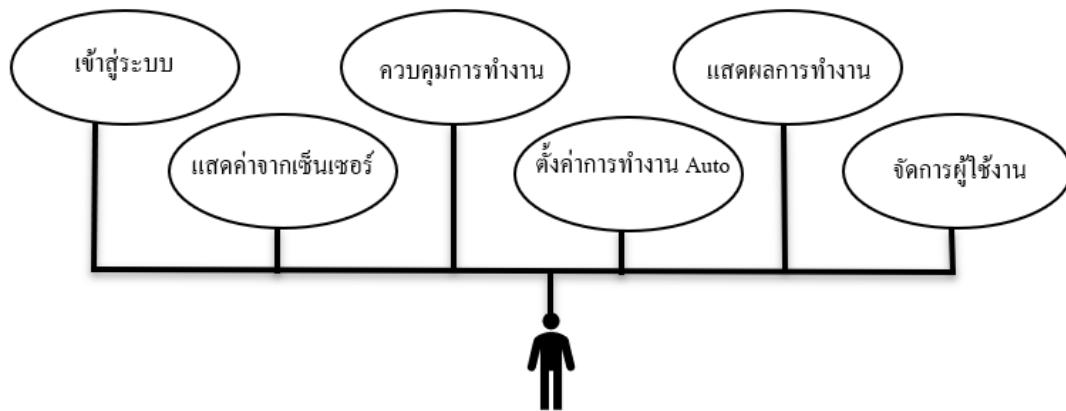
3.3.5 Entity timers ประกอบด้วย Attribute

- timer_id (Pk) หมายเลขเวลา เป็น Primary key
- created_date เวลาที่บันทึก
- timer_name ชื่อเวลา
- start_time เวลาที่เริ่ม
- start_duration_sec ระยะเวลา
- light_value ค่าสถานะ เปิด/ปิด ไฟ
- pump_value ค่าสถานะ เปิด/ปิด ปั๊ม
- aircon_value ค่าอุณหภูมิ

- temp_value ค่าสถานะ เปิด/ปิด ทำความเย็น
- fan_value ค่าสถานะ เปิด/ปิด พัดลมระบาย
- start_repeating_sec จำนวนครั้งที่ให้ทำงานซ้ำ
- controller_id เลขอุปกรณ์

3.3.6 การออกแบบเว็บแอปพลิเคชัน

ในการออกแบบ เว็บแอปพลิเคชัน จะมี 2 สถานะผู้ใช้งาน คือ แอดมิน และ ผู้ใช้งานทั่วไป โดยที่ผู้ใช้งานทั่วไปนั้นจะเข้าใช้งานได้ก็ต่อเมื่อได้รับการจัดการเข้าสู่ระบบจากแอดมินก่อน โดย สิทธิในการควบคุมและแสดงผลของแอดมินและผู้ใช้งานทั่วไปจะเหมือนกัน แตกต่างกันที่สิทธิในการจัดการผู้ใช้งานเท่านั้น



รูปที่ 3.7 Use case Diagram web application

จาก Use case Diagram ประกอบด้วย 5 User Interface ดังนี้

1) หน้าเข้าสู่ระบบ



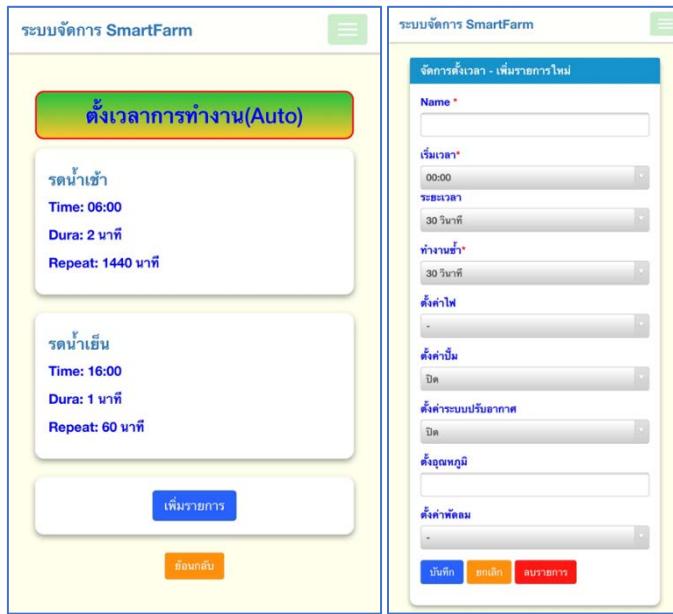
รูปที่ 3.8 ภาพแสดงหน้าจอเข้าสู่ระบบ

2) หน้าแสดงค่าและควบคุมการทำงาน



รูปที่ 3.9 ภาพแสดงหน้าจอควบคุมการทำงาน

3) หน้าการตั้งค่าการทำงาน



รูปที่ 3.10 ภาพแสดงหน้าจอตั้งค่าการทำงาน

4) หน้าแสดงผลการทำงาน



รูปที่ 3.11 ภาพแสดงหน้าจอแสดงผลการทำงาน

5) หน้าแสดงการจัดการผู้ใช้งาน

Username	ชื่อ-สกุล	Email	ประเภท
admin	admin	sarut1245@gmail.com	ผู้ดูแลระบบ
tester	Tester AAA	test@test.com	ผู้ใช้งาน
sarut	aaaa bbbbb	60015044@kmitl.ac.th	ผู้ดูแลระบบ

แสดง 1 ถึง 3 จาก 3 ผล

ก่อนหน้า 1 ถัดไป

ระบบจัดการ SmartFarm

จัดการผู้ใช้งานระบบ - เพิ่มรายการใหม่

Username *

ชื่อ-สกุล *

อีเมล *

รหัสผ่าน *

โทรศัพท์ *

ห้อง *

วันเดือนปี *

Email *

ประเภท *

บันทึก ลบเมือง ลบรายการ

รูปที่ 3.12 ภาพแสดงหน้าจอการจัดการผู้ใช้งาน

บทที่ 4

การทดลองและผลการทดลอง

บทนี้เป็นการทดลองต่าง ๆ ทั้งส่วนตัวของเรื่อง และส่วน Web Application โดยได้ทำการทดลองดังนี้

1. สร้างตัวจำลองโรงเรือนแบบปิด
2. การทดลองการเข้าใช้งาน web application
3. การทดลองการส่งข้อมูลและสั่งงานผ่านอินเตอร์เน็ตด้วย web application
4. การทดลองการอ่านข้อมูลต่าง ๆ ย้อนหลังจาก web application
5. การทดลองการรับน้ำและปรับอุณหภูมิแบบอัตโนมัติ
6. การทดลองเพาะปลูกผักสดในตู้โรงเรือนจำลอง

4.1 สร้างตัวจำลองโรงเรือนแบบปิด

4.1.1 วัตถุประสงค์

เพื่อทดลองโรงเรือนสำหรับปลูกพืชพกที่สามารถควบคุมอุณหภูมิ แสงสว่างและความชื้นต่าง ๆ ภายในโรงเรือนได้

4.1.2 วิธีการดำเนิน

สร้างโรงเรือนจำลองจากแผ่นพลาสติก ขนาด 100 x 60 x 60 เซนติเมตร มีการบุผนังภายในด้วยโฟมหนึ่งชั้น ภายในติดตั้ง หลอดไฟ LED สำหรับเพาะปลูก เชิงเซอร์ DHT22 เชิงเซอร์วัดความชื้นในดิน พัดลมดูดอากาศ ระบบระดับน้ำ, น้ำ และระบบพ่นหมอกเพื่อ控อุณหภูมิในโรงเรือน ภายในนอกมีจอ LCD สำหรับดูค่าต่าง ๆ หน้าตู้ และมีบอร์ดควบคุม



รูปที่ 4.1 ตู้จำลองโรงเรือนแบบปิด

4.1.3 ผลการทำงาน

ได้ตู้โรงเรือนจำลองแบบปิดเพื่อสำหรับปลูกพืชผัก ภายในติดตั้งอุปกรณ์ที่ใช้สำหรับเพาะปลูก

4.2 ทดลองการเข้าใช้งาน Web application

4.2.1 วัตถุประสงค์

เพื่อทดสอบระบบการจัดการผู้ดูแล และผู้ใช้งาน web application ให้สามารถเข้าใช้งานได้เฉพาะผู้ที่มี User ที่ผู้ดูแลเพิ่มไว้ หากไม่มี user จะไม่สามารถเข้าใช้งานได้

4.2.2 วิธีการดำเนิน

เข้าใช้งานใน User ของผู้ดูแลเพื่อเข้าไปเพิ่มผู้ใช้งาน หลังจากนี้ให้ทดลองเข้าสู่ระบบโดย User ที่เพิ่งทำการสร้างขึ้นมาใหม่ เมื่อเข้าได้แล้วทำงานของจากระบบ แล้วเข้า login ใหม่ โดยใช้ User ที่ยังไม่ได้มีการเพิ่มเข้าใหม่ในระบบ

ระบบจัดการ SmartFarm ผู้ดูแล

admin (ผู้ดูแลระบบ) +

บันทึกการแก้ไข : ผู้ดูแลระบบ

Username * : tanaporn

รหัสผ่าน * : ****

ยืนยันรหัสผ่าน * : ****

ชื่อ * : tanaporn

นามสกุล * : sukrasom

โทรศัพท์ * : 60015623

โทรศัพท์มือถือ * : 0922320956

Email * : 60015623@kmitl.ac.th

ตำแหน่ง * : ผู้ใช้งาน

บันทึก **แก้ไข** **ลบ**

รูปที่ 4.2 การเพิ่มUserในส่วนของผู้ดูแล

4.2.3 ผลการทำงาน

หากจะทำงานเพิ่ม User ต้องเพิ่มจากผู้ดูแลเท่านั้น ผู้ใช้งานทั่วไปไม่สามารถเพิ่ม User ใหม่ได้ และหากไม่มี User หรือกรอกข้อมูลผิดก็ไม่สามารถเข้าใช้งาน web application ได้

ระบบจัดการ SmartFarm ผู้ดูแล

เข้าสู่ระบบ

ไม่สามารถเข้าสู่ระบบได้ เนื่องจาก รหัสผู้ใช้งานหรือรหัสผ่าน หรือ รหัสผ่านถูกต้อง

ผู้ใช้งาน:

Username:

รหัสผ่าน:

Password:

เข้าสู่ระบบ

รูปที่ 4.3 กรณ์ไม่มีUser หรือ กรอกข้อมูลผิด

4.3 ทดลองการส่งข้อมูลและสั่งงานผ่านอินเตอร์เน็ตด้วย Web application

4.3.1 วัตถุประสงค์

เพื่อทดลองการรับส่งข้อมูลและการสั่งงานผ่านอินเตอร์เน็ตด้วย Web application ให้สามารถควบคุมการทำงานและอ่านค่าต่าง ๆ จากเซ็นเซอร์ได้

4.3.2 วิธีการดำเนิน

ทดลองควบคุมการเปิด-ปิดไฟLED อ่านค่าอุณหภูมิและความชื้นในอากาศ ผ่านWeb application ให้สามารถควบคุมผ่านคอมพิวเตอร์หรือสมาร์ทโฟน แม้อยู่คนละวงLANกัน



รูปที่ 4.4 หน้าweb application ที่ใช้ควบคุม ผ่านสมาร์ทโฟน

4.3.3 ผลการทำงาน

สามารถควบคุมการเปิด-ปิดไฟLED และอ่านค่าอุณหภูมิและความชื้นในอากาศได้



รูปที่ 4.5 สถานะหน้าจอเปลี่ยนตามที่ควบคุมผ่าน wed application

4.4 ทดลองการอ่านข้อมูลต่าง ๆ ย้อนหลังจาก web application

4.4.1 วัตถุประสงค์

เพื่ออ่านข้อมูลต่าง ๆ ที่ใช้ในการเพาะปลูก สำหรับไว้วิเคราะห์และวางแผนการเพาะปลูกพืชชนิดนั้นในภายหลัง หรือคูปัจจัยต่าง ๆ ที่มีผลต่อการเจริญเติบโต

4.4.2 วิธีการดำเนิน

เข้าweb application เลือกหน้ารายงาน จากนั้นจะมีข้อมูลของวันต่าง ๆ แสดงขึ้นมา

4.4.3 ผลการทำงาน

สามารถอ่านข้อมูลต่าง ๆ ย้อนหลังจาก web application ได้

เวลา	ความชื้น	ความชื้นอากาศ	อุณหภูมิ	ความชื้น
22:33	0	0	0	0
22:44	0	75.8	31.2	13.3333
22:44	0	74.4	31.2	11.6667
22:44	0	74.4	31.2	13.3333
22:44	0	74.4	31.2	13.3333
22:44	0	74.1	31.1	13.3333
22:44	0	74.1	31.1	13.3333
22:45	0	74.1	31	13.3333
22:45	0	74.2	31	13.3333
22:45	0	74.1	31	13.3333
22:45	0	74.4	30.9	13.3333
22:45	0	74.5	30.9	13.3333
22:45	0	74.5	30.9	13.3333
22:46	0	74.7	30.8	13.3333

รูปที่ 4.6 ข้อมูลต่าง ๆ ย้อนหลังบน web application

4.5 ทดลองการรดน้ำและปรับอุณหภูมิแบบอัตโนมัติ

4.5.1 วัตถุประสงค์

เพื่อทดสอบการรดน้ำตามค่าความชื้นในดินที่ตั้งไว้ และทดสอบการควบคุมอุณหภูมิและความชื้นในอากาศตามที่ได้ตั้งค่าไว้

4.5.2 วิธีการดำเนิน

เข้าweb application เลือกหน้าที่ใช้ควบคุม ตั้งค่าความชื้นในดิน ความชื้นในอากาศ และอุณหภูมิ จากนั้นดูการทำงาน เมื่ออุณหภูมิสูงกว่าที่กำหนดระบบพ่นหมอกต้องทำงาน หากความชื้น

ในคืนต่อว่าที่กำหนดระบบจะทำงาน และถ้าความชื้นในอากาศสูงกว่าที่กำหนดพัดลมจะทำงาน



รูปที่ 4.7 หน้าเมนูต่างๆ บน web application

4.5.3 ผลการทำงาน

สามารถทำงานได้ตามที่กำหนดไว้



รูปที่ 4.8 หน้ารายการที่กำหนดการทำงานต่างๆ ไว้

4.6 ทดลองเพาะปลูกผักสดในตู้โรงเรือนจำลอง

4.6.1 วัตถุประสงค์

เพื่อทดสอบประสิทธิภาพของระบบ ว่าสามารถช่วยในการเพาะปลูกได้ สามารถควบคุมปัจจัยต่าง ๆ เพื่อให้พืชสามารถเจริญเติบโตได้

4.6.2 วิธีการดำเนิน

ทำการเพาะปลูกผักสด ตามที่ได้ศึกษา ตั้งค่าการควบคุมความชื้นในดิน ความชื้นในอากาศ อุณหภูมิ จากนั้นอยด์ติดตามผลการเจริญ โตของผักสด



รูปที่ 4.9 การปลูกผักในตู้จำลองโรงเรือน

4.6.3 ผลการทำงาน

การเจริญเติบโตของขึ้นอุ่นอยู่กับหลายปัจจัย โดยในการทดลองครั้งแรก เมล็ดผักที่ทำการปลูกมีบางเมล็ดที่ไม่萌芽 ซึ่งอาจเกิดจากคุณภาพของเมล็ด จึงได้ทำการเพาะเมล็ดใหม่ และเลือกเมล็ดที่มีรากงอกลงดินแทน ซึ่งกำลังติดตามผลการเจริญเติบโตต่อไป

บทที่ 5

บทสรุปและวิจารณ์

5.1 ผลการดำเนินงาน

- 1) สร้างตู้จำลองโรงเรือนแบบปิดให้เหมาะสมกับการวางแผนอุปกรณ์ในการควบคุม
- 2) สร้างวงจรอิเล็กทรอนิกส์ที่ใช้สำหรับคอนโทรลระบบ
- 3) พัฒนาโปรแกรมให้สามารถควบคุมและแสดงผลหน้าจอได้
- 4) พัฒนาเว็บแอพพลิเคชันที่สามารถเปิดได้ในเว็บเบราว์เซอร์

5.2 ปัญหา อุปสรรค และแนวทางการแก้ไข

5.2.1 ปัญหา อุปสรรค

- 1) มีปัญหาในการสร้างตู้จำลองโรงเรือนแบบปิด ซึ่งเกิดการร้าวซึมเมื่อ/runnning และการระบายน้ำที่ไม่ดี
- 2) มีปัญหาเกี่ยวกับตู้จำลองโรงเรือนแบบปิด ซึ่งเราไม่สามารถควบคุมอุณหภูมิได้ตามที่ต้องการ รวมถึงการแจ้งเตือนการเติมน้ำของอุปกรณ์ที่ไม่สามารถคาดเดาได้
- 3) การออกแบบระบบในแต่ละส่วนมีความละเอียดมากจึงทำให้เกิดความล่าช้า และทำให้บางครั้งต้องทำการออกแบบใหม่ในบางส่วน
- 4) ไม่มีความติดในการเขียนเว็บไซต์ ทำให้ทำงานได้ล่าช้า
- 5) คุณภาพของเมล็ดที่นำมาเพาะปลูกไม่ดี ทำให้มีการไม่งอกหรือไม่โต

5.2.2 แนวทางการแก้ไข

- 1) ทำการยิงกาวซิลิโคลนรอบ ๆ รอยต่อภายใน เจาะรูระบายน้ำให้ใหญ่ขึ้น และทำการอุ่นพื้นไปทางรูระบายน้ำเล็กน้อย เพื่อให้น้ำไหลไปในทิศทางของรูระบายน้ำ
- 2) เพิ่มจำนวนไฟฟ้าที่ทำเป็นจำนวนมากให้หนาขึ้น เลือกซื้ออุปกรณ์จากโรงงานที่นำเข้ามา หรือมีประวัติ
- 3) ออกแบบให้รอบครอบ พยายามทำตามขอบเขตที่กำหนด
- 4) ศึกษาเกี่ยวกับข้อมูลเพิ่มเติมทั้งจากในอินเตอร์เน็ต หนังสือ และปรึกษากับผู้เชี่ยวชาญที่สนใจ
- 5) ศึกษาวิธีการเพาะปลูกอย่างละเอียด และปรับเปลี่ยนที่ชื้อเมล็ดพันธุ์ที่นำมาเพาะปลูก

5.3 แนวทางการพัฒนา

- 1) พัฒนาฟังก์ชันการใช้งานใหม่ๆ เพิ่มเติมต่อไปอีกในอนาคต เช่น การสั่งงานแบบอัตโนมัติ, การแจ้งเตือนความเสี่ยหายของอุปกรณ์
- 2) มีการติดตั้งระบบทำความสะอาดเย็นแบบ compressor เพื่อให้สามารถควบคุมอุณหภูมิที่ต่ำๆ ได้ เพื่อที่จะสามารถเพาะปลูกพืชเมืองหนาวได้
- 3) พัฒนาเว็บแอพพลิเคชันให้สามารถใช้งานได้อย่างเสถียรมากขึ้น และปรับปรุงให้มีความน่าใช้หน้าสนใจ
- 4) นำระบบไปประยุกต์ใช้ในโรงพยาบาลใหญ่

บรรณานุกรม

จีราวด วารินทร์. 2562. พัฒนาเว็บไซต์สมัยใหม่ด้วย HTML5 JavaScript + CSS3 ฉบับสมบูรณ์.

กรุงเทพฯ : บริษัท รีไวว่า จำกัด

ปัญชา ประศีลະเตสัง. 2562. สร้างเว็บไซต์แบบ Resposive ด้วย Bootstrap ร่วมกับ CSS และ

JavaScript.

กรุงเทพฯ : บริษัท ชีเอ็คยูคชั่น จำกัด (มหาชน).

ศุภชัย สมพาณิช. 2561. พัฒนาเว็บแอพด้วย JavaScript แบบ MEAN Stack.

กรุงเทพ : บริษัท สวัสดิ์ ไอที จำกัด.

รศ.ชาญชัย ศุภอรรถกร. 2561. สร้างเว็บแอพพลิกชัน PHP MySQL+AJAX jQuery ฉบับสมบูรณ์.

กรุงเทพฯ : บริษัท รีไวว่า จำกัด.

ธีระพล ลีมศรีทชา. 2562. ออกแบบเว็บไซต์ รูปแบบ และสถาปัตกรรม ด้วย PHP.

กรุงเทพฯ : บริษัท ชีเอ็คยูคชั่น จำกัด (มหาชน).

สุพจน์ สง่ากุล. 2562. การเขียนโปรแกรมภาษา Python.

กรุงเทพฯ : บริษัท รีไวว่า จำกัด.

โฉตพันธ์ หล่อเลิศสุนทร และฐิตพันธ์ หล่อเลิศสุนทร. 2559. คู่มือเรียน เขียนโปรแกรม Python.

กรุงเทพฯ : คอร์ฟังก์ชั่น

ผศ.ดร.ชัชชัย ฤทธิ์. 2562. IOT: สถาปัตยกรรมการสื่อสาร = Internet of Thing.

กรุงเทพฯ : บริษัท ชีเอ็คยูคชั่น จำกัด (มหาชน).

ดร.กอบเกียรติ สารอุบล. 2561. พัฒนา IOT บนแพรตฟอร์ม Arduino และ Raspberry Pi.

กรุงเทพฯ : อินเตอร์มิเดีย

วรากรณ์ บุญเพ็ชร์ และสุจารี บุญสิทธิ์. 2559. “ระบบสนับสนุนการประเมินผลการเรียนรู้ของผู้เรียน. “ปริญญา妮พนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ บัณฑิต วิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ชุษิพงษ์ ศรีเกยม, สมมาตร เพียรชำนาญ และอนุชิต จึงศรีพิมณ. 2551. “ระบบตรวจสอบสถานะและจำนวนสินค้าของตู้หยดเหรียญอัตโนมัติ. “ปริญญา妮พนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.