



รายงานสหกิจศึกษาฉบับสมบูรณ์

การแสดงผลข้อมูลการวัดพลังงานระยะไกลผ่านเว็บ
Web-Based Remote Monitoring for Power Telemetry

นางสาวพิมพ์พรรณ ปรีदानนท์

หลักสูตรวิศวกรรมอัตโนมัติ
ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561



รายงานสหกิจศึกษาฉบับสมบูรณ์

การแสดงผลข้อมูลการวัดพลังงานระยะไกลผ่านเว็บ
Web-Based Remote Monitoring for Power Telemetry

นางสาวพิมพ์พรรณ ปรีदानนท์

หลักสูตรวิศวกรรมอัตโนมัติ
ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา	การแสดงผลข้อมูลการวัดพลังงานระยะไกลผ่านเว็บ
ชื่อ-สกุล นักศึกษา	นางสาวพิมพ์พรรณ ปรีदानนท์ รหัสนักศึกษา 58010895
หลักสูตร	วิศวกรรมอัตโนมัติ
คณะ	วิศวกรรมศาสตร์
ชื่อ-สกุล อาจารย์นิเทศ	ผศ.ดร. ธีรวัฒน์ เทพมณี ผศ.ดร.กฤษณ์ เสมอพิทักษ์
ชื่อ-สกุล ผู้นิเทศงาน	คุณสิริชัย วงศ์ยุทธนาพงศ์
ชื่อสถานประกอบการ	บริษัท สเกต้า ออโตเมชั่น จำกัด

บทคัดย่อ

การวัดพลังงานระยะไกลที่ได้ศึกษาถูกออกแบบให้มีการแสดงผลข้อมูลการใช้ไฟฟ้าแบบระยะใกล้และไกล ซึ่งประกอบด้วยเพาเวอร์มิเตอร์รุ่น EASTRON SMART X96-3 ที่ใช้สำหรับการวัดค่าพารามิเตอร์ทางไฟฟ้า บอร์ดคอมพิวเตอร์รุ่น ASUS Tinker ที่ใช้สำหรับการบันทึกค่าที่วัดได้ในฐานข้อมูล ส่วนแสดงผลด้วย LabVIEW ที่ใช้สำหรับการแสดงผลข้อมูลการวัดพลังงานระยะไกลแบบเรียลไทม์ และเว็บที่ใช้สำหรับการแสดงผลข้อมูลการวัดพลังงานระยะไกลแบบเรียลไทม์และแบบข้อมูลย้อนหลังใน 1 วัน จุดมุ่งหมายของโครงการสหกิจศึกษานี้คือการออกแบบและสร้างเว็บสำหรับแสดงผลข้อมูลการวัดพลังงานระยะไกลที่ได้ศึกษา โดยเว็บที่สร้างขึ้นจะส่งคำสั่งร้องขอทุก 1 วินาทีเพื่อเรียกค่าพารามิเตอร์ที่ถูกบันทึกไว้ในฐานข้อมูลจากบอร์ด Tinker และเว็บจะนำค่าที่ได้รับนั้นมาแสดงบนหน้าเว็บทั้ง 4 หน้าที่สร้างขึ้นในรูปแบบตาราง ในกรณีของการแสดงผลข้อมูลย้อนหลังใน 1 วัน ผู้ใช้งานสามารถเลือกวันที่ของข้อมูลที่ต้องการแสดงได้ในรูปแบบ YYYY-MM-DD จากนั้นเว็บจะทำการส่งคำสั่งร้องขอเพื่อเรียกข้อมูลในวันที่ต้องการ หลังสิ้นสุดการดาวน์โหลดข้อมูลย้อนหลังค่าพารามิเตอร์จะถูกนำมาแสดงผลในรูปแบบตาราง จากผลการทดสอบการแสดงผลแบบเรียลไทม์และการแสดงผลข้อมูลย้อนหลังใน 1 วัน ยืนยันได้ว่าเว็บที่สร้างขึ้นสามารถแสดงผลข้อมูลการวัดพลังงานได้ตามต้องการ

คำสำคัญ: เว็บ, ระยะไกล, การแสดงผล, เพาเวอร์มิเตอร์, การวัดระยะไกล

Cooperative Project Title: Web-Based Remote Monitoring for Power Telemetry
Student: Miss Pimpran Preedanont Student ID 58010895
Program: Automation Engineering
Faculty: Engineering
Advisors: Asst. Prof. Dr. Teerawat Thepmanee
Asst. Prof. Dr. Krit Smerpitak
Mentor: Mr. Sirichai Wongyutthanapong
Company: Scada Automation Company Limited

ABSTRACT

The studied power telemetry is designed to locally and remotely monitor electricity usage. It consists of a power meter modeled EASTRON SMART X96-3 for measuring the electrical parameters, a board computer modeled ASUS Tinker for recording the measured values in the database, a LabVIEW-based monitoring for locally displaying the recorded parameters in real time, and a web-based monitoring for remotely displaying the recorded parameters in real time as well as the historian data in a day. The aim of this project is to design and implement the web-based monitoring for the studied telemetry. The created web sends the request every second to get the recorded parameters from the database in the Tinker board. The received values are then shown on the created web pages in table format. In case of 1-day historian data display, the user can select the day in format of YYYY-MM-DD, then the created web sends the request to get the recorded data during the specified date. After downloading the required historian data, the electrical values are shown in table format. Experimental test results verify that both real-time parameters and 1-day historian data from the database can be monitored the created web pages.

Keywords: Web, Remote, Monitoring, Power Meter, Telemetry

กิตติกรรมประกาศ

โครงการสหกิจศึกษาฉบับสมบูรณ์ฉบับนี้จะสำเร็จลุล่วงไปไม่ได้หากไม่ได้รับความอนุเคราะห์จากบริษัทสเกต้า ออโตเมชัน จำกัด ที่เปิดโอกาสให้ผู้จัดทำได้เข้าร่วมโครงการสหกิจศึกษาเพื่อเรียนรู้งานและวิธีการทำงานในสถานประกอบการจริง ผู้จัดทำขอขอบคุณคุณรณน สติชัยปัญญาพันธ์ เจ้าของบริษัทผู้ซึ่งมอบหมายงานและให้คำปรึกษา ตลอดจนคำแนะนำต่าง ๆ ในทุกเรื่อง และขอขอบพระคุณคุณสิริชัย วงศ์ยุทธนาพงศ์ ผู้นิเทศน์งานที่คอยให้คำแนะนำและช่วยเหลือจนโครงการสหกิจศึกษาฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี รวมถึงพนักงานในบริษัททุกท่านที่ให้ความช่วยเหลือมากมายในทุก ๆ ด้าน

ขอขอบพระคุณ ผศ.ดร. ธีรวัฒน์ เทพมณี และผศ.ดร.กฤษณ์ เสมอพิทักษ์ ที่มอบโอกาสให้ผู้จัดทำได้ออกไปศึกษาค้นคว้าและร่วมทำโครงการกับสถานประกอบการ รวมถึงทุกความช่วยเหลือและคำแนะนำที่ได้รับตลอดการปฏิบัติงานในระยะเวลาสี่เดือน และขอขอบพระคุณ รศ.ดร. อัมพวัน จุลเสรีวงศ์ ที่ให้ความรู้และช่วยตรวจสอบรูปเล่มตลอดจนให้คำแนะนำที่ดีจนกระทั่งรายงานสหกิจศึกษาเล่มนี้เสร็จครบถ้วนสมบูรณ์

ขอขอบพระคุณอาจารย์ประจำหลักสูตรวิศวกรรมอัตโนมัติทุกท่านที่คอยสั่งสอนและถ่ายทอดความรู้ทั้งทางทฤษฎีและปฏิบัติ ซึ่งทุกองค์ความรู้ที่ได้รับล้วนเป็นประโยชน์แก่การนำไปประยุกต์ใช้ในการทำงานจริง ตลอดจนสามารถนำไปใช้เป็นประสบการณ์เพื่อดำเนินชีวิตในฐานะวิศวกรได้ต่อไป

ขอขอบคุณทุกความรู้ที่ได้รับจากหนังสือทุกเล่ม เอกสารอ้างอิงทุกฉบับ ที่ผู้จัดทำได้นำมาใช้ อ้างอิงเพื่อทำโครงการสหกิจศึกษาฉบับนี้ ท้ายสุดผู้จัดทำขอขอบพระคุณครอบครัวที่คอยให้การสนับสนุนและเป็นกำลังใจที่ดีให้ผู้จัดทำตลอดมา

พิมพ์พรรณ ปรีदानนท์

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 แผนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 แนวคิดและหลักการที่เกี่ยวข้อง.....	4
2.1 กล่าวนำ	4
2.2 แนวคิดของการวัดข้อมูลพลังงานระยะไกล.....	4
2.2.1 ภาพรวมของการวัดข้อมูลพลังงานระยะไกล.....	4
2.2.2 เพาเวอร์มิเตอร์.....	5
2.2.3 ระบบไฟฟ้าสามเฟส.....	7
2.2.4 บอร์ด ASUS Tinker	8
2.3 องค์ประกอบของเว็บ.....	10
2.4 ซอฟต์แวร์ที่เกี่ยวข้อง.....	11
2.4.1 Visual studio code.....	11
2.4.2 XAMPP.....	12
2.4.3 SQLite.....	13
2.5 ภาษาโปรแกรมสำหรับการสร้างเว็บ	14
2.5.1 ภาษา HTML (Hyper Text Markup Language).....	14
2.5.2 ภาษา CSS (Cascading Style Sheets)	16
2.5.3 ภาษา PHP (Hypertext Preprocessor).....	19
2.5.4 ภาษา JavaScript	21
2.6 โพรโตคอล HTTP	25

สารบัญ (ต่อ)

หน้า

บทที่ 3 การแสดงผลข้อมูลระยะไกลที่นำเสนอ	28
3.1 กล่าวนำ	28
3.2 การออกแบบและสร้างเว็บสำหรับการแสดงผลที่นำเสนอ.....	28
3.2.1 การออกแบบเว็บ	28
3.2.2 การเขียนโปรแกรมเพื่อสร้างเว็บ	29
3.3 การเขียนโปรแกรมส่วนแสดงผลและสร้างฐานข้อมูลจำลองเพื่อทดสอบระบบ	35
3.3.1 สร้างฐานข้อมูลจำลองและเชื่อมต่อฐานข้อมูล.....	35
3.3.2 การทำ AJAX เพื่อให้แสดงผลแบบเรียลไทม์	39
3.4 การเชื่อมต่อกับฐานข้อมูลของบอร์ด ASUS TINKER และนำมาแสดงผลแบบเรียลไทม์	41
3.5 การแสดงค่าข้อมูลย้อนหลังโดยสามารถเลือกวันที่ได้	47
3.5.1 การป้อน Input วันที่เพื่อเลือกหน้า web service	48
3.5.2 Data tables	49
บทที่ 4 ผลการทดสอบโครงการ	52
4.1 กล่าวนำ	52
4.2 การแสดงค่าแบบเรียลไทม์.....	52
4.3 การแสดงข้อมูลย้อนหลังโดยสามารถเลือกวันที่ที่ต้องการแสดงค่าย้อนหลังได้	53
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	57
5.1 สรุปผลการดำเนินงาน	57
5.2 ปัญหาและวิธีการแก้ไข.....	57
5.2.1 ปัญหาที่พบ.....	57
5.2.2 วิธีการแก้ไข	57
5.3 ข้อเสนอแนะ	58
เอกสารอ้างอิง	59

สารบัญตาราง

ตารางที่	หน้า
1.1 ตารางแสดงหัวข้องาน และระยะเวลาการดำเนินงาน	2
2.1 อุปกรณ์ที่ติดตั้งในระบบ	5
3.1 ตัวแปรและพารามิเตอร์	43



สารบัญรูป

รูปที่	หน้า
2.1 ภาพรวมของการวัดข้อมูลพลังงานระยะไกล.....	4
2.2 Analog Power Meter และ Digital Power Meter.....	5
2.3 ตัวอย่างค่าพารามิเตอร์ที่เพาเวอร์มิเตอร์สามารถวัดได้	6
2.4 สามค่าหลักที่จำเป็นต่อการวิเคราะห์กำลังไฟฟ้า	7
2.5 ระบบไฟฟ้าสามเฟส.....	7
2.6 Logo ของบอร์ด ASUS Tinker.....	8
2.7 องค์ประกอบของบอร์ด ASUS Tinker	8
2.8 หน่วยประมวลผล ARM แบบ Quad-core.....	9
2.9 พอร์ตเชื่อมต่อต่าง ๆ ของบอร์ด ASUS Tinker	10
2.10 ส่วน front-end และ back-end ของเว็บ	11
2.11 Visual studio code software	12
2.12 Logo xampp software.....	12
2.13 Logo apache software	13
2.14 โปรแกรม SQLite3.....	13
2.15 โปรแกรม DB Browser Portable ส่วนเสริมสำหรับ SQLite3.....	14
2.16 โครงสร้าง HTML	14
2.17 การใช้ <title> และการแสดงค่า.....	15
2.18 การสร้างตารางโดยใช้ <table>.....	16
2.19 การเปรียบเทียบระหว่างเว็บแอปพลิเคชันแบบดั้งเดิมกับแบบที่ใช้ AJAX	21
2.20 การเปรียบเทียบการสื่อสารระหว่างเว็บแอปพลิเคชันแบบดั้งเดิมกับแบบที่ใช้ AJAX	22
2.21 Logo ของ jQuery.....	24
2.22 ตัวอย่างของ Data table	25
2.23 ตัวอย่างของโค้ดในเว็บ Data tables	25
3.1 การออกแบบหน้าเว็บด้วยโปรแกรม PowerPoint	29
3.2 ตัวอย่างการเขียนโค้ด HTML เบื้องต้น	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.3 การแสดงผลของโค้ด HTML ส่วนของ Head	30
3.4 การแสดงผลของโค้ด HTML ส่วนของ Body.....	30
3.5 การแสดงผลเมื่อใส่ภาพพื้นหลังแล้ว.....	31
3.6 เว็บไซต์ https://fonts.google.com/	31
3.7 การใช้งานฟอนต์จากเว็บไซต์ https://fonts.google.com/	32
3.8 การเปลี่ยนรูปแบบฟอนต์ในส่วน HTML.....	32
3.9 การเปลี่ยนรูปแบบฟอนต์ในส่วน CSS.....	32
3.10 การแสดงผลเมื่อเปลี่ยนรูปแบบฟอนต์บนหน้าเว็บ	33
3.11 การใช้ class ใน HTML เพื่อตกแต่งแต่ละส่วน.....	33
3.12 การใช้ class ใน CSS เพื่อตกแต่งแต่ละส่วน	33
3.13 ตัวอย่างการแสดงผลเมื่อสร้างเมนูบาร์เสร็จสิ้น.....	35
3.14 การ Configuration Apache เพื่อใช้งาน SQLite.....	36
3.15 โค้ดเชื่อมต่อกับฐานข้อมูลของ SQLite.....	37
3.16 หน้าเว็บทดสอบแสดงค่าเมื่อเชื่อมต่อฐานข้อมูลแล้ว.....	39
3.17 Web service ของบอร์ด ASUS Tinker	42
3.18 การแสดงผลเมนู Total.....	47
3.19 ข้อมูลย้อนหลังบน Web service ของบอร์ด ASUS Tinker	48
3.20 Library ของ jQuery ที่โหลดมาจากเว็บ DataTables	49
4.1 หน้าเว็บที่ 1 Total แสดงค่าพารามิเตอร์แบบเรียลไทม์.....	52
4.2 หน้าเว็บที่ 2 ค่าปริมาณพลังงานไฟฟ้า แสดงค่าพารามิเตอร์แบบเรียลไทม์	53
4.3 หน้าเว็บที่ 3 Parameter แสดงค่าพารามิเตอร์แบบเรียลไทม์.....	53
4.4 ช่องค้นหาวันที่ในเว็บ.....	54
4.5 การกรอกวันที่ลงในช่องค้นหา	54
4.6 เมื่อกด Find จะขึ้น loading ระหว่างเรียกข้อมูล.....	54
4.7 เมื่อข้อมูลย้อนหลังโหลดเสร็จแล้ว	55

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.8 ข้อมูลย้อนหลังแบบเลือกวันที่ในตาราง.....	55
4.9 ฟังก์ชันตารางข้อมูลย้อนหลัง.....	55



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ทางบริษัทเสกต้า ออโตเมชัน จำกัด รับทำโครงการต่าง ๆ มากมาย รวมถึงโครงการ Solar Rooftop SCADA ที่เป็นหนึ่งในโครงการที่ทางบริษัทรับทำระบบให้อยู่บ่อยครั้ง ในการทำโครงการ Solar Rooftop SCADA นั้นจะต้องมีการตรวจสอบว่าโรงงานอุตสาหกรรมมีการใช้ไฟฟ้ามากที่สุดในช่วงเวลาใดของวัน ซึ่งไม่สามารถตรวจสอบได้จากมิเตอร์ไฟฟ้าของทางการไฟฟ้า จึงจำเป็นต้องนำเพาเวอร์มิเตอร์ (Power Meter) ไปติดตั้งเพื่อเก็บค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าที่ใช้ในแต่ละวันและนำมาวิเคราะห์ต่อไป

จากการนำเพาเวอร์มิเตอร์เข้าไปติดตั้งที่ตู้ไฟของโรงงาน ทางบริษัทได้พบปัญหาเรื่องความไม่สะดวกและไม่ปลอดภัยเป็นอย่างมาก เนื่องจากการเก็บข้อมูลเพื่อนำมาทำการวิเคราะห์จะต้องเก็บข้อมูลพลังงานไฟฟ้ามากกว่าห้าวันขึ้นไป และตัวเพาเวอร์มิเตอร์เองนั้นจะอ่านค่าได้เพียงแค่ที่หน้าจอแสดงผลบนตัวอุปกรณ์ที่ติดตั้งอยู่ในตู้ไฟเท่านั้น เมื่อเกิดความผิดพลาดจากการติดตั้งเพาเวอร์มิเตอร์หรือความผิดพลาดจากตัวอุปกรณ์เอง จะทำให้เกิดความสูญเปล่าของเวลาเป็นอย่างมาก เนื่องจากโรงงานอุตสาหกรรมส่วนใหญ่ตั้งอยู่ในพื้นที่ต่างจังหวัด จึงไม่อาจล่วงรู้และแก้ไขความผิดพลาดเหล่านั้นได้ทัน และทำให้ต้องสูญเสียเวลาไปโดยเปล่าประโยชน์ รวมถึงการเดินทางเพื่อไปเก็บข้อมูลที่โรงงานก็ทำให้เกิดค่าใช้จ่ายมากขึ้นอีกด้วย

บริษัทเสกต้า ออโตเมชัน จำกัดจึงมอบหมายให้ผู้จัดทำนำเพาเวอร์มิเตอร์มาพัฒนาในส่วนของการแสดงผลระยะไกลเพิ่มเติมเพื่อแก้ปัญหาข้างต้น โดยนำบอร์ด ASUS Tinker มาเป็นตัวกลางในการเก็บข้อมูลและส่งออกข้อมูลสู่โลกอินเทอร์เน็ต โดยใช้โปรโตคอล Modbus ในการสื่อสารกันระหว่างบอร์ด ASUS Tinker และเพาเวอร์มิเตอร์ ค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าจะถูกบันทึกลง SD Card บนบอร์ด ASUS Tinker ส่งต่อข้อมูลผ่านโปรโตคอล HTTP เพื่อนำไปแสดงผลบนเว็บและโปรแกรม LabVIEW ต่อไป

1.2 วัตถุประสงค์ของโครงการ

ออกแบบและสร้างเว็บให้สามารถแสดงค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าสำหรับการวัดข้อมูลพลังงานระยะไกลได้ โดยมีการแสดงผลแบบเรียลไทม์ (Real-Time) และข้อมูลย้อนหลัง (Historian data) ซึ่งผู้ใช้งานสามารถกำหนดช่วงเวลาของข้อมูลย้อนหลังที่ต้องการได้

1.3 ขอบเขตของโครงการ

- เขียนเว็บด้วยภาษา HTML, CSS และ JavaScript สำหรับแสดงผลข้อมูลค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าจากระยะไกล โดยเว็บแสดงผลนี้สร้างมาเพื่ออำนวยความสะดวกในการเข้าถึงข้อมูลค่าพลังงานให้แก่ผู้ใช้งาน ซึ่งเป็นเจ้าของโรงงานหรือผู้ที่เกี่ยวข้องเท่านั้น
- เว็บที่สร้างขึ้นสามารถเข้าถึงผ่านเว็บเบราว์เซอร์ได้
- ค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าที่วัดได้จากเพาเวอร์มิเตอร์รุ่น EASTRON SMART X96-3 และเก็บบันทึกลงใน SD Card ของบอร์ด ASUS Tinker จะถูกนำมาแสดงผลบนหน้าเว็บ โดยแบ่งข้อมูลที่นำมาแสดงออกเป็น 3 กลุ่ม ได้แก่ กลุ่มพลังงานรวมของระบบไฟฟ้าสามเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มค่าพารามิเตอร์พื้นฐาน เช่น กระแสไฟฟ้าและค่าแรงดันไฟฟ้าในแต่ละเฟส และกลุ่มค่าพลังงานในแต่ละเฟส

4. เว็บสำหรับแสดงผลจะแบ่งออกเป็น 4 पेจ คือ पेจแสดงข้อมูลแบบเรียลไทม์ของกลุ่มพลังงานรวมของระบบไฟฟ้าสามเฟส, पेจแสดงข้อมูลแบบเรียลไทม์ของกลุ่มค่าพารามิเตอร์พื้นฐาน, पेจแสดงข้อมูลแบบเรียลไทม์ของกลุ่มค่าพลังงานในแต่ละเฟส และ पेจแสดงข้อมูลย้อนหลัง

1.4 แผนการดำเนินงาน

1. ศึกษาค้นคว้าการเขียนโปรแกรมด้วยภาษา HTML, CSS และ JavaScript รวมถึงการใช้งานโปรโตคอล HTTP
2. สอบถามรายละเอียดของงาน ฟังก์ชันการทำงานของเว็บที่ต้องการ ข้อมูลการวัดพลังงานที่ต้องการนำขึ้นแสดงบนเว็บจากผู้มอบหมายงาน จากนั้นนำข้อมูลทั้งหมดมาออกแบบการจัดวางบนหน้าเว็บแต่ละ पेจ และเขียนโปรแกรมเพื่อสร้างเว็บตามโครงร่างที่ได้ออกแบบเอาไว้
3. สร้างฐานข้อมูลจำลองเพื่อทดสอบการแสดงผลแบบเรียลไทม์ของเว็บ
4. เชื่อมต่อเว็บกับฐานข้อมูลของบอร์ด ASUS Tinker ด้วยโปรโตคอล HTTP
5. เขียนโปรแกรมสำหรับการแสดงข้อมูลย้อนหลังโดยสามารถรอกวันที่เพื่อเลือกวันที่ของข้อมูลที่เก็บบันทึกไว้

ตารางที่ 1.1 ตารางแสดงหัวข้องาน และระยะเวลาการดำเนินงาน

	ส.ค. 2561	ก.ย. 2561	ต.ค. 2561	พ.ย. 2561
1. ศึกษาค้นคว้าการเขียนโปรแกรมด้วยภาษา HTML, CSS และ JavaScript รวมถึงการใช้งานโปรโตคอล HTTP				
2. ออกแบบและเขียนโปรแกรมสร้างเว็บ				
3. สร้างฐานข้อมูลจำลองเพื่อทดสอบการแสดงผลแบบเรียลไทม์ของเว็บ				
4. เชื่อมต่อเว็บกับฐานข้อมูลของบอร์ด ASUS Tinker ด้วยโปรโตคอล HTTP				
5. เขียนโปรแกรมสำหรับการแสดงข้อมูลย้อนหลังโดยสามารถรอกวันที่เพื่อเลือกวันที่ของข้อมูลที่เก็บบันทึกไว้				

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถตรวจสอบข้อมูลการวัดพลังงานแบบเรียลไทม์และเรียกดูข้อมูลย้อนหลังผ่านเว็บจากระยะไกลได้
2. ลดค่าใช้จ่ายจากการเดินทาง ลดเปอร์เซ็นต์ความผิดพลาดที่เกิดขึ้น และเพิ่มประสิทธิภาพในการทำงานให้สามารถทำงานได้สะดวกและรวดเร็วมากยิ่งขึ้น



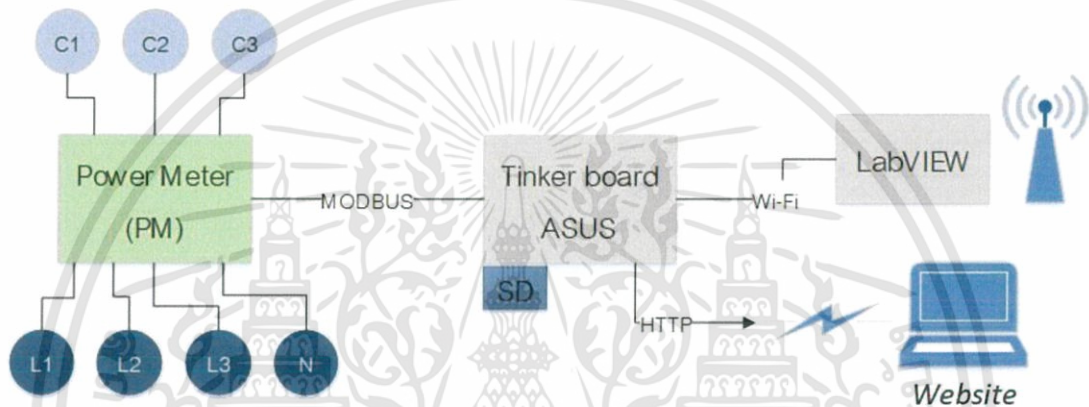
บทที่ 2 แนวคิดและหลักการที่เกี่ยวข้อง

2.1 กล่าวนำ

ในการดำเนินการของโครงการ จะต้องมีการศึกษาแนวคิดและหลักการที่เกี่ยวข้องกับโครงการเพื่อใช้เป็นข้อมูลรองรับในโครงการ โดยสามารถสรุปเนื้อหาที่เกี่ยวข้อง ได้ดังต่อไปนี้

2.2 แนวคิดของการวัดข้อมูลพลังงานระยะไกล

2.2.1 ภาพรวมของการวัดข้อมูลพลังงานระยะไกล



รูปที่ 2.1 ภาพรวมของการวัดข้อมูลพลังงานระยะไกล

การวัดข้อมูลพลังงานระยะไกล จะใช้เพาเวอร์มิเตอร์วัดไฟฟ้าสามเฟส รุ่น EASTRON SMART X96-3 มาสร้างฟังก์ชันการใช้งานเพิ่มเติม โดยนำบอร์ด ASUS Tinker มาเป็นตัวกลางในการรับส่งและเก็บบันทึกข้อมูลค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าและใช้โปรโตคอล Modbus เป็นสื่อกลางการติดต่อสื่อสารระหว่างบอร์ด ASUS Tinker และเพาเวอร์มิเตอร์

เมื่อบอร์ด ASUS Tinker อ่านค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าด้วยโปรโตคอล Modbus และนำข้อมูลมาบันทึกลงฐานข้อมูลบน SD Card เรียบร้อยแล้วบอร์ด ASUS Tinker จะทำหน้าที่เป็น Server ในการแชร์ข้อมูลจากฐานข้อมูลออกมาในรูปแบบ Web service เพื่อให้เว็บและโปรแกรม LabVIEW สามารถดึงข้อมูลออกมาใช้ได้ โดยติดต่อสื่อสารกันผ่านโปรโตคอล HTTP ซึ่งเป็นการติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันแบบไร้สาย

ตารางที่ 2.1 อุปกรณ์ที่ติดตั้งในระบบ

อุปกรณ์	หน้าที่
1. Power Meter รุ่น EASTRON SMART X96-3	เครื่องตรวจวัดพลังงานไฟฟ้า
2. Tinker Board ASUS	Controller ที่ใช้เป็น Server และตัวกลางในการรับส่งข้อมูลที่วัดได้จากเพาเวอร์มิเตอร์
3. Labtop หรือ Computer	อุปกรณ์สำหรับเรียกแสดงเว็บ

2.2.2 เพาเวอร์มิเตอร์

เพาเวอร์มิเตอร์ คือ อุปกรณ์แสดง "ค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้า" เช่น แรงดัน, กระแส, กำลังงานไฟฟ้าจริง, กำลังงานไฟฟ้ารีแอกทีฟ และ Harmonic เป็นต้น เพื่อให้ทราบถึงค่าทางไฟฟ้าในกระบวนการผลิตและการใช้พลังงานไฟฟ้าได้ โดยส่วนใหญ่แล้วในภาคอุตสาหกรรม จะนำเพาเวอร์มิเตอร์ ไปใช้ในการควบคุมหรือปรับปรุงการใช้พลังงานไฟฟ้า เพื่อให้เกิดประสิทธิภาพในการทำงานได้อย่างเต็มที่ อีกทั้งยังเป็นการช่วยจัดการพลังงาน ซึ่งเป็นไปตามมาตรฐานของ ISO 50001

โดยเพาเวอร์มิเตอร์นั้น สามารถแบ่งออกได้เป็น 2 แบบ คือ Analog Power Meter (แบบเข็ม) และ Digital Power Meter (แบบหน้าจอดีจิตอล)



รูปที่ 2.2 Analog Power Meter และ Digital Power Meter

เพาเวอร์มิเตอร์สามารถวัดค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าได้เป็นร้อยค่าตามคู่มือการใช้งาน ดังนั้นขึ้นอยู่กับผู้ใช้งานที่ต้องการเลือกแสดงค่าใดบ้าง

Address (Register)	SDM630MT Input Register Parameter		Modbus Protocol Start Address Hex		3 0	3 0	1 0
	Description	Units	Hi Byte	Lo Byte	4 W	3 W	2 W
30001	Phase 1 line to neutral volts.	Volts	00	00	✓	X	✓
30003	Phase 2 line to neutral volts.	Volts	00	02	✓	X	X
30005	Phase 3 line to neutral volts.	Volts	00	04	✓	X	X
30007	Phase 1 current.	Amps	00	06	✓	✓	✓
30009	Phase 2 current.	Amps	00	08	✓	✓	X
30011	Phase 3 current.	Amps	00	0A	✓	✓	X
30013	Phase 1 power.	Watts	00	0C	✓	X	✓
30015	Phase 2 power.	Watts	00	0E	✓	X	✓
30017	Phase 3 power.	Watts	00	10	✓	X	X
30019	Phase 1 volt amps.	VA	00	12	✓	X	✓
30021	Phase 2 volt amps.	VA	00	14	✓	X	X
30023	Phase 3 volt amps.	VA	00	16	✓	X	X
30025	Phase 1 volt amps reactive.	VAr	00	18	✓	X	✓
30027	Phase 2 volt amps reactive.	VAr	00	1A	✓	X	X

รูปที่ 2.3 ตัวอย่างค่าพารามิเตอร์ที่เฟาเวอร์มิเตอร์สามารถวัดได้

การวิเคราะห์กำลังไฟฟ้า จะต้องวิเคราะห์ค่า 3 ประเภทหลัก ๆ ดังนี้

■ Active Power (P)

คือ กำลังไฟฟ้าที่ใช้จริง เกิดจากโหลดความต้านทาน มีหน่วยเป็น วัตต์ (W) หรือกิโลวัตต์ (KW)

คำนวณได้จากสมการ $P = V \times I \times \text{Cos}(\text{zeta})$

■ Reactive Power (Q)

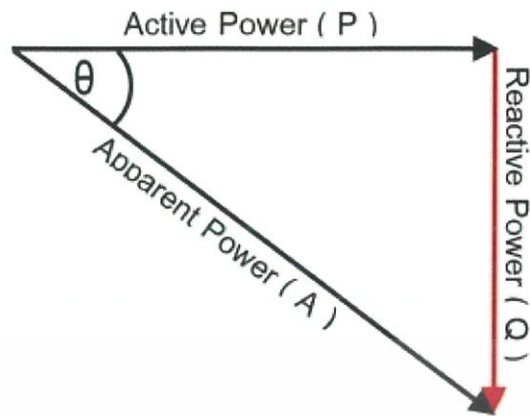
คือ กำลังไฟฟ้าที่สูญเสีย เกิดจากโหลดตัวเหนี่ยวนำและตัวเก็บประจุ มีหน่วยเป็น วาร์ (VAR) หรือกิโลวาร์ (kVAR)

คำนวณได้จากสมการ $Q = V \times A \times \text{Sin}(\text{zeta})$

■ Apparent Power (A)

คือกำลังไฟฟ้าที่ปรากฏ (Input) หรือผลรวมทางเวกเตอร์ของไฟฟ้าที่ใช้จริง และกำลังไฟฟ้าที่สูญเสีย มีหน่วยเป็นโวลต์ แอมแปร์ (VA) หรือกิโลโวลต์ แอมแปร์ (kVA)

คำนวณได้จากสมการ $Q = V \times A \times \text{Sin}(\text{zeta})$

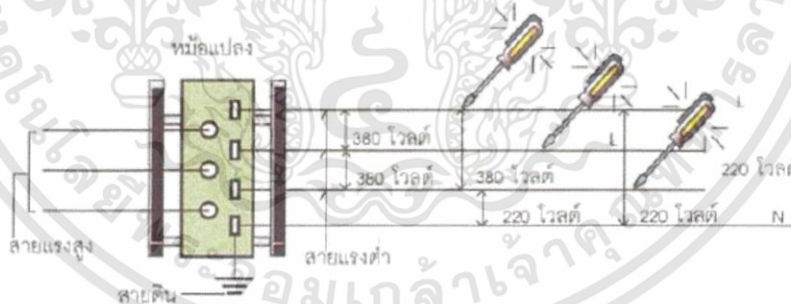


รูปที่ 2.4 สามค่าหลักที่จำเป็นต่อการวิเคราะห์กำลังไฟฟ้า

2.2.3 ระบบไฟฟ้าสามเฟส

ระบบไฟฟ้า 3 เฟส ระบบไฟฟ้า 3 เฟสเป็นระบบไฟฟ้ากระแสสลับ 3 เฟส (3-phase) 4 สาย แรงดัน 380 โวลต์ ความถี่ 50 เฮิรตซ์ โดยที่ 3 สายจะเป็นสายที่มี กระแสไฟฟ้าไหลผ่าน

ระบบไฟ 3 เฟส



รูปที่ 2.5 ระบบไฟฟ้าสามเฟส

โดยทั่วไประบบไฟฟ้า 3 เฟสเป็นระบบที่ไฟฟ้าที่ใช้กับเครื่องจักรต่าง ๆ ใน โรงงาน อุตสาหกรรมเป็นส่วนใหญ่ เพราะเครื่องจักรเหล่านี้มักมีขนาดใหญ่จึงต้องการแรงดันไฟฟ้าที่สูง ไฟฟ้า ระบบนี้ไม่สามารถนำมาใช้กับระบบแสงสว่างหรืออุปกรณ์ไฟฟ้าตามบ้านได้โดยตรง มาถึงตรงนี้ หลายคนคงจะสงสัยว่าเมื่อระบบไฟฟ้า 3 เฟสไม่สามารถนำมาใช้กับอุปกรณ์ไฟฟ้าต่าง ๆ ตามบ้านได้ โดยตรงแล้วจะเอามาแนะนำกันเพื่ออะไร ข้อสงสัยนี้สามารถอธิบายได้โดยไม่ยาก กล่าวคือ การนำ ระบบไฟฟ้า 3 เฟสเข้ามาใช้ในบ้านนั้นมิได้เป็นการใช้ไฟฟ้าทั้ง 3 เฟสกับอุปกรณ์ไฟฟ้าแต่ละชิ้น โดยตรง แต่เป็นการนำไฟฟ้า 3 เฟสนั้นมาแบ่งแยกให้เป็นระบบไฟฟ้า 1 เฟส 3 ชุด แล้วกระจายไป ตามจุดต่าง ๆ ที่มีการใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

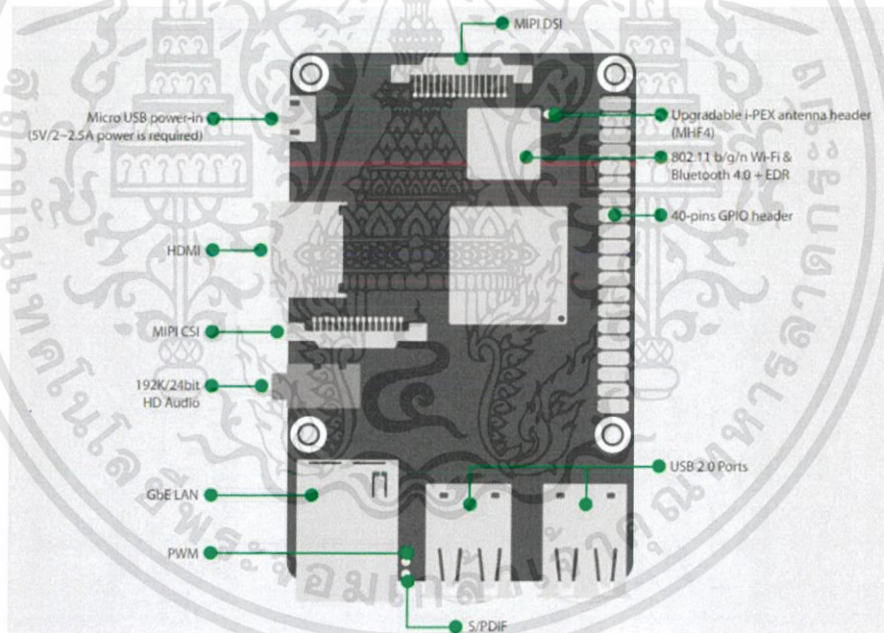
การกระจายจุดของการใช้งานเช่นนี้ทำให้ไฟฟ้าแต่ละเฟสไม่ถูกใช้งานมาก ถือเป็นทางเลือกในการใช้ไฟฟ้า ทำให้ประหยัดค่าไฟฟ้า เพราะการคิดอัตราค่าใช้ไฟฟ้าซึ่งมีหน่วยเป็น กิโลวัตต์-ชั่วโมงจะคิดเป็นอัตราก้าวหน้า กล่าวคือยิ่งมีการใช้ไฟฟ้ามากก็จะยิ่งเสียค่าไฟฟ้าในอัตราที่สูงขึ้น ฉะนั้นการกระจาย การใช้ไฟฟ้าออกเป็น 3 ส่วนจากระบบไฟฟ้าที่นำเข้า 3 เฟสดังกล่าว จึงทำให้การใช้ไฟฟ้าในแต่ละส่วนหรือแต่ละเฟสน้อยลง จึงไม่ต้องเสียค่าไฟฟ้าในอัตราที่สูง

2.2.4 บอร์ด ASUS Tinker



รูปที่ 2.6 Logo ของบอร์ด ASUS Tinker

บอร์ด ASUS Tinker เป็น Single Board Computer (SBC) ในรูปแบบฟอร์มแฟคเตอร์แบบ ultra-small ที่ให้คุณได้สัมผัสกับประสิทธิภาพที่เหนือกว่า



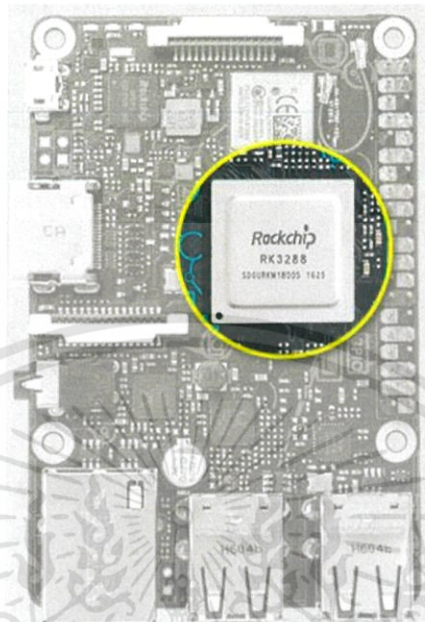
รูปที่ 2.7 องค์ประกอบของบอร์ด ASUS Tinker

Features & Functionality

- **Class-Leading Performance**

ด้วยหน่วยประมวลผล ARM แบบ Quad-core ที่ทรงประสิทธิภาพและทันสมัย Rockchip RK3288 ทำให้บอร์ด ASUS Tinker สามารถมอบประสิทธิภาพที่เหนือกว่าเมื่อเทียบกับ SBC ทั่วไป เราได้คำนึงถึงความต้องการสร้างโปรเจกต์ที่เพิ่มมากขึ้นบอร์ด ASUS Tinker จึงมาพร้อมกับหน่วยความจำ LPDDR3 2GB แบบ Dual-channel และยังมีพร้อม

กับอินเตอร์เฟซ SD 3.0 เพื่อให้ประสิทธิภาพความเร็วการอ่านและเขียนลงใน microSD สำหรับระบบปฏิบัติการ แอปพลิเคชัน และการเก็บข้อมูล



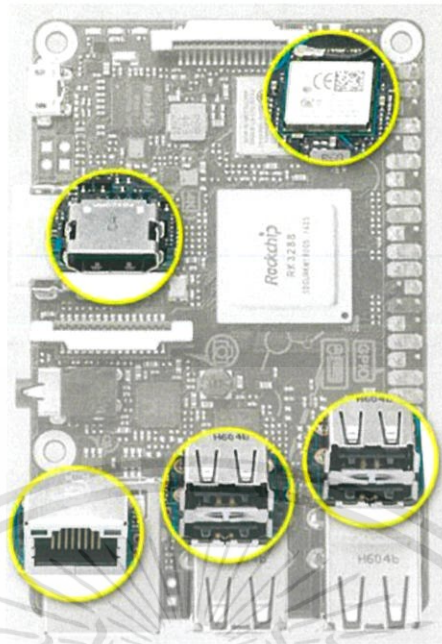
รูปที่ 2.8 หน่วยประมวลผล ARM แบบ Quad-core

- **Maker-Friendly, with IoT Connectivity**

บอร์ด ASUS Tinker มาพร้อมกับตัวเลือกการเชื่อมต่อสำหรับเมคเกอร์และผู้ชื่นชอบงานอดิเรก รวมถึงอินเตอร์เฟซ 40-pin GPIO นอกจากนี้ยังมาพร้อมกับการเชื่อมต่อ HD MIPI 2 พอร์ตเพื่อการแสดงผล HD และกล้อง HD

บอร์ด ASUS Tinker ยังมีพอร์ต LAN Gbit เพื่อการโอนถ่ายข้อมูลที่เหนือกว่าเหมาะแก่การเป็นศูนย์กลางระบบเครือข่ายและเป็นอุปกรณ์จัดเก็บข้อมูลบนระบบเครือข่าย อีกทั้งพอร์ต LAN บนบอร์ด ASUS Tinker ยังได้รับทรัพยากรพิเศษ เพื่อให้สามารถโอนถ่ายข้อมูลที่มีประสิทธิภาพ ภาคควบคุม Wi-Fi และ Bluetooth ได้รับการป้องกันสัญญาณรบกวนด้วยแผ่นโลหะ เพื่อให้คุณได้รับประสิทธิภาพสูงสุด และยังมีช่องเชื่อมต่อสายอากาศ IPEX อีกด้วย

นอกจากการเชื่อมต่อข้างต้นแล้วบอร์ด ASUS Tinker ยังมีพอร์ต HDMI แบบมาตรฐานเพื่อเชื่อมต่อกับ TV, จอภาพ และการแสดงผลรูปแบบอื่น พร้อมพอร์ต USB 2.0 4 พอร์ตเพื่อการเชื่อมต่ออุปกรณ์เสริมที่ครบครัน



รูปที่ 2.9 พอร์ตเชื่อมต่อต่าง ๆ ของบอร์ด ASUS Tinker

2.3 องค์ประกอบของเว็บ

ก่อนการสร้างเว็บจะต้องทราบว่าในการทำงานของเว็บนั้นแบ่งการทำงานออกเป็นสองส่วน นั่นคือส่วน front-end กับ back-end ซึ่งเป็นส่วนหนึ่งของระบบ CMS (Content Management System) หรือ ระบบจัดการเนื้อหาของเว็บ

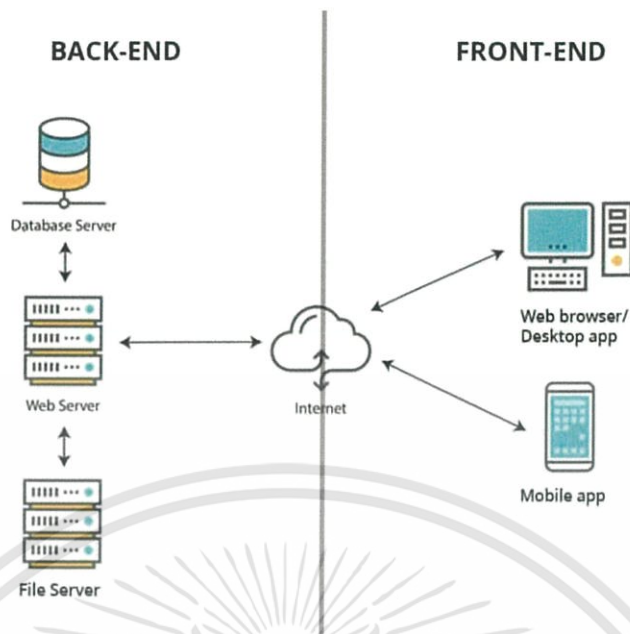
➤ ส่วน front-end

เรียกกันโดยทั่วไปว่าหน้าบ้าน หรือเป็นส่วนติดต่อผู้ใช้ (User interface) ไม่ว่าจะเป็น โสมเพจ หน้าเว็บ เนื้อหาต่าง ๆ รูปภาพ ลิงก์ เป็นต้น เป็นส่วนที่ User ทั่วไปสามารถเห็นและเข้ามาใช้งานได้ของเว็บ

➤ ส่วน back-end

เรียกกันโดยทั่วไปว่า หลังบ้าน หรือระบบจัดการเว็บ เช่น จัดการฐานข้อมูล โครงสร้างเว็บ การเขียนโค้ดควบคุม XML, text file, JAVA, PHP, C#, C++ เป็นต้น จะมีไว้สำหรับ admin หรือผู้ที่ได้รับอนุญาต เพื่อทำการ เพิ่ม ลบ แก้ไข เปลี่ยนแปลงเว็บ

ส่วนของ Back-end มีความสำคัญเป็นอย่างยิ่ง ไม่ว่าจะเป็นการรักษาความปลอดภัยของข้อมูล ทั้ง Username password ข้อมูลเว็บต่าง ๆ หาก Back-end มีการ Update ก็ต้องทำการ Backup ข้อมูลเดิมเก็บไว้ และตรวจสอบให้ดีก่อนทำการ Update รวมไปถึงการทำงานของเว็บ ความเร็วในการแสดงผล



รูปที่ 2.10 ส่วน front-end และ back-end ของเว็บ

Front-end กับ Back-end จะมีความสำคัญต่อกันเป็นอย่างมาก หาก Front-end แสดงผลช้าหรือเกิดข้อผิดพลาด ส่วนของ Back-end ก็ต้องทำการแก้ไขปัญหา หรือถ้าหากว่า Back-end ไม่ทำการปรับปรุงเว็บอย่างสม่ำเสมอ จะส่งผลต่อเว็บเป็นอย่างมาก ไม่ว่าจะเป็น เว็บล่าช้า ขาดความน่าสนใจ ผู้เยี่ยมชมลดลง ก่อให้เกิดผลกระทบหลาย ๆ ด้าน

ดังนั้นการสร้างเว็บจึงต้องมีทั้งส่วนของ Front-end และ Back-end ซึ่งต้องเขียนแยกกันเพื่อความง่ายและสะดวกในการทำงาน

แต่ในโครงการนี้มีหน้าเว็บเพียงเพื่อติดต่อกับผู้ใช้งาน คือให้แสดงผลค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้า รวมถึงแสดงผลย้อนหลังโดยสามารถเลือกแสดงผลตามวันที่ได้ ดังนั้นการเขียนเว็บที่มีรายละเอียดไม่เยอะมากจึงไม่จำเป็นต้องแยก Front-end กับ Back-end ออกจากกัน

2.4 ซอฟต์แวร์ที่เกี่ยวข้อง

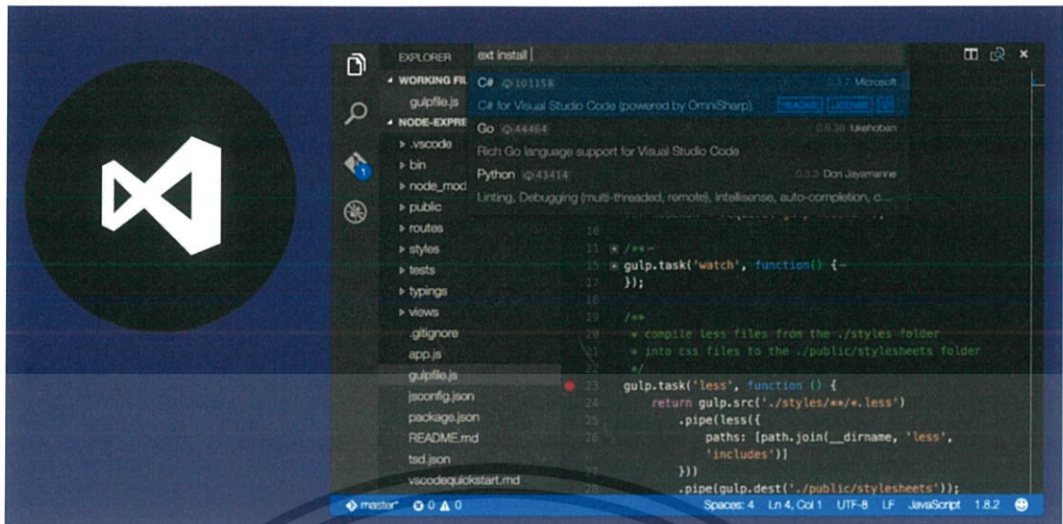
2.4.1 Visual studio code

Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพ

ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาย ไม่ว่าจะเป็น

- 1.การเปิดใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP หรือ Go
- 2.Themes
- 3.Debugger
- 4.Commands เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 2.11 Visual studio code software

ความแตกต่างระหว่าง VSCode และ Visual Studio คือ

- VSCode ได้ทำการตัดในส่วนของ GUI designer ออกไป เหลือแต่เพียงตัว Editor เท่านั้น จึงทำให้ตัวโปรแกรมนั้นค่อนข้างเบากว่า Visual Studio เป็นอย่างมาก
- VSCode สามารถนำมาใช้งานได้ฟรี รองรับการทำงานข้ามแพลตฟอร์ม

2.4.2 XAMPP



รูปที่ 2.12 Logo xampp software

Xampp คือโปรแกรม WebServer จำลอง โดยใช้คอมพิวเตอร์ส่วนบุคคลของเราให้ทำงานเป็น WebServer ดังนั้นคอมพิวเตอร์ส่วนบุคคลจะกลายเป็นทั้ง Client และ Server ในเครื่องเดียวกัน จึงไม่มีความจำเป็นในการเชื่อมต่ออินเทอร์เน็ตในการเรียกเว็บไซต์ขึ้นมาแสดง เป็นโปรแกรมที่มอบความสะดวกแก่ผู้สร้างเว็บไซต์และต้องการทดสอบเว็บไซต์นั้นได้ทุกที่ทุกเวลาตามต้องการ

XAMPP ประกอบด้วย Apache, PHP, MySQL, PHP MyAdmin, Perl ซึ่งเป็นโปรแกรมพื้นฐานที่รองรับการทำงาน CMS ซึ่งเป็นชุดโปรแกรมสำหรับออกแบบเว็บไซต์ที่ได้รับความนิยมในปัจจุบัน ไฟล์สำหรับติดตั้ง xampp นั้นอาจมีขนาดใหญ่พอสมควร เนื่องจาก มีชุดควบคุมการทำงานที่ช่วยให้การปรับแต่งส่วนต่าง ๆ ง่ายขึ้น

- Apache



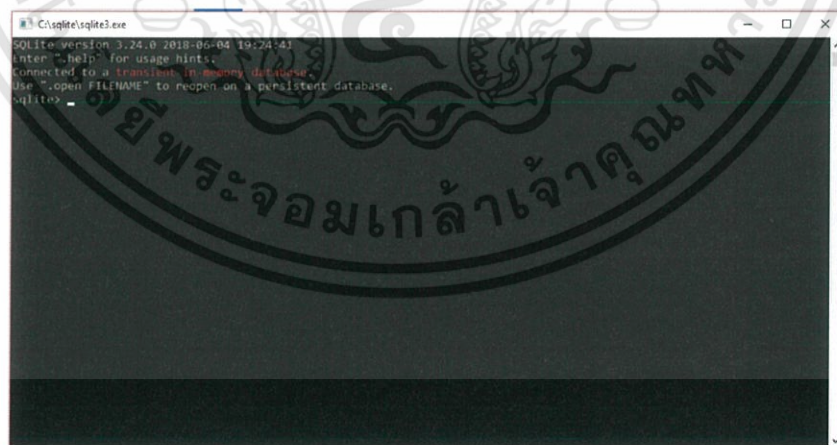
รูปที่ 2.13 Logo apache software

Apache คือ WebServer ที่พัฒนามาจาก HTTPD WebServer ซึ่ง Apache นี้จะทำหน้าที่ในการจัดเก็บ Homepage และส่ง Homepage ไปยัง Browser ที่มีการเรียก ซึ่งปัจจุบันจัดได้ว่าเป็น WebServer ที่มีความน่าเชื่อถือมาก เนื่องจากเป็นที่นิยมใช้กันทั่วโลก อีกทั้ง Apache ยังเป็นซอฟต์แวร์แบบ Open source ที่เปิดให้บุคคลทั่วไปสามารถเข้ามาร่วมพัฒนาส่วนต่าง ๆ ของ Apache ได้ ซึ่งทำให้เกิดเป็นโมดูลที่เกิดประโยชน์มากมาย เช่น mod_perl, mod_python หรือ mod_php และทำงานร่วมกับภาษาอื่นได้ แทนที่จะเป็นเพียงเซิร์ฟเวอร์ที่ให้บริการเพียงแค่ HTML อย่างเดียว

นอกจากนี้ Apache เองยังมีความสามารถอื่น ๆ ด้วย เช่นการยืนยันตัวบุคคล (mod_auth, mod_access, mod_digest) หรือเพิ่มความปลอดภัยในการสื่อสารผ่านโปรโตคอล HTTPS (mod_ssl) และยังมีโมดูลอื่น ๆ ที่ได้รับความนิยมใช้อีกด้วย

2.4.3 SQLite

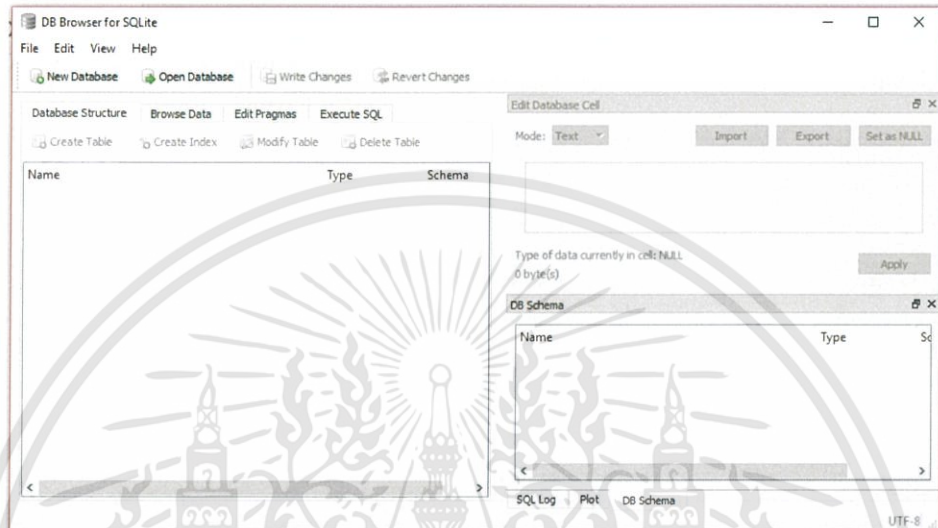
SQLite เป็นโปรแกรมฐานข้อมูลที่มีขนาดเล็กมาก (ไม่ถึง 1MB) เก็บฐานข้อมูลเป็นไฟล์โดยไม่จำเป็นต้องมีเซิร์ฟเวอร์ ทำให้ถูกใช้ในหลาย ๆ โปรแกรมหรือถูกติดตั้งลงในอุปกรณ์พกพาหลายชนิด ๆ เช่น iPhone, Android เพื่อใช้ในการเก็บข้อมูล



รูปที่ 2.14 โปรแกรม SQLite3

SQLite เป็นระบบจัดการฐานข้อมูล (Database Management System) หรือที่เรียกว่า ดีบีเอ็มเอส (DBMS) เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางในระบบติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ภายในฐานข้อมูล ซึ่งต่างจากระบบแฟ้มข้อมูลที่หน้าที่เหล่านี้จะเป็นหน้าที่ของโปรแกรมเมอร์ ในการติดต่อเอกสารเป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติไหนไปใช้ประโยชน์ด้านการค้า

กับข้อมูลในฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่มดีเอ็มแอล (DML) หรือ ดีดีแอล (DDL) หรือจะด้วยโปรแกรมต่าง ๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูลจะถูกตีบีเอ็มเอสนำมาแปล (คอมไพล์) เป็นการปฏิบัติการ (Operation) ต่าง ๆ ภายใต้อคำสั่งนั้น ๆ เพื่อนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูลต่อไป สำหรับส่วนการทำงานต่าง ๆ ภายในดีบีเอ็มเอสที่ทำหน้าที่แปลคำสั่งไปเป็นการปฏิบัติการต่าง ๆ กับข้อมูลนั้น



รูปที่ 2.15 โปรแกรม DB Browser Portable ส่วนเสริมสำหรับ SQLite3

2.5 ภาษาโปรแกรมสำหรับการสร้างเว็บ

2.5.1 ภาษา HTML (Hyper Text Markup Language)

เป็นภาษาหลักที่ใช้ในการสร้างโฮมเพจ จะประกอบไปด้วย tag เปิด และ tag ปิด ยกตัวอย่าง เช่น <html>...</html>

```
<HTML>
<HEAD>
<TITLE>.....</TITLE>
</HEAD>
<BODY>
.....
.....
</BODY>
</HTML>
```

รูปที่ 2.16 โครงสร้าง HTML

โดยทั่วไปแล้วภาษา HTML นั้นจะประกอบไปด้วย Tag สำคัญ ๆ ดังนี้

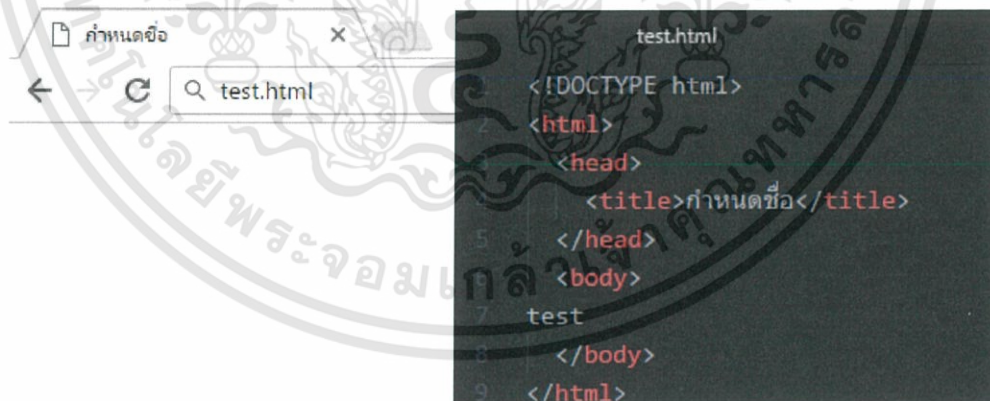
- <html>.....</html> เป็นแท็กคำสั่งในการเริ่มต้นของภาษา HTML
- <head>.....</head> เป็นแท็กคำสั่งที่จะบอกคุณลักษณะ หรือคุณสมบัติของหน้าเว็บ
- <title>.....</title> เป็นแท็กคำสั่งแสดงชื่อของหน้าเว็บ

- <body>.....</body> เป็นส่วนที่ใช้แสดงเนื้อหาของหน้าเว็บ ไม่ว่าจะเป็น ข้อมูลต่าง ๆ รูปภาพ วิดีโอ ตาราง หรือสื่อข้อมูลอื่น ๆ

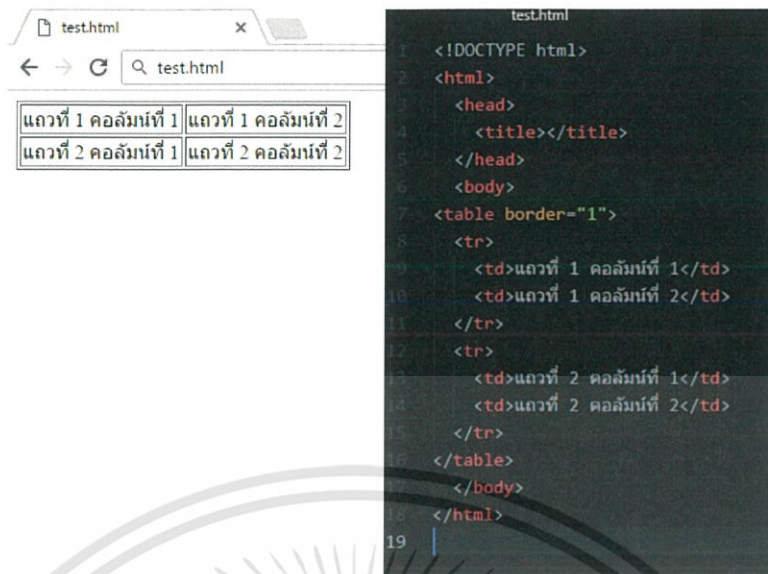
ในการเขียนคำสั่งภาษา HTML สามารถเขียน ด้วยตัวอักษร เล็กหรือใหญ่ ทั้งหมดหรือเขียนคละกันได้ เช่น <HTML> หรือ <Html> หรือ <html> ซึ่งจะให้ผลเหมือนกัน

คำสั่งการใช้งานเบื้องต้นในภาษาHTML ซึ่งเป็นคำสั่งที่ใช้กันบ่อย ๆ

<p><html> -แท็กเริ่มต้น</p> <p><!-...-> -ข้อความด้านในจะแสดงในหน้าโค้ด แต่ไม่แสดงในหน้าเว็บ</p> <p><head> -แท็กบอกคุณสมบัติ</p> <p><title> -แท็กแสดงชื่อหน้าเว็บ</p> <p><body> -แท็กแสดงเนื้อหา</p> <p>ข้อความที่แสดง</p> <p> -แท็กการเชื่อมโยง</p> <p> -แท็กแสดงรูปภาพ</p> <p>
 -แท็กขึ้นบรรทัดใหม่</p> <p><table> -แท็กสร้างตาราง</p> <p><tr> -แท็กสร้างแถว</p> <p><td> -แท็กสร้างคอลัมน์</p>	<p><hr> -แท็กเส้นคั่น</p> <p> -แท็กตัวอักษรหนา</p> <p><i> -แท็กตัวอักษรเอียง</p> <p><u> -แท็กตัวอักษรขีดเส้นใต้</p> <p><marquee> -แท็กตัวอักษรเลื่อน</p> <p> -แท็กรูปแบบอักษร</p> <p><h1> ถึง <h6> -แท็กหัวข้อเรื่อง</p> <p><input> -แท็กเพิ่มข้อมูล</p> <p><label> -แท็กข้อความ</p> <p><select> -แท็กลิสต์เมนู</p> <p><dd> -แท็กเยื้องหน้า</p> <p><p> -แท็กย่อหน้า</p> <p><center> -แท็กการจัดกึ่งกลาง</p>
--	--



รูปที่ 2.17 การใช้ <title> และการแสดงค่า



รูปที่ 2.18 การสร้างตารางโดยใช้ <table>

2.5.2 ภาษา CSS (Cascading Style Sheets)

เป็นภาษาที่มีรูปแบบการเขียน Syntax ที่มีความเฉพาะ ใช้ในการกำหนดรูปแบบเอกสารเว็บ หรือตกแต่งเอกสาร HTML ให้มีหน้าตา สี สัน ตัวอักษร ตำแหน่ง เส้นขอบ พื้นหลัง ระยะห่าง ตามต้องการ โดยการกำหนดคุณสมบัติให้กับ Element ต่าง ๆ ของ HTML

ประโยชน์ของ CSS

1. ช่วยจัดรูปแบบการแสดงผล ช่วยลดขนาดของไฟล์ HTML ทำให้ประมวลได้อย่างรวดเร็ว และประหยัดพื้นที่จัดเก็บ
2. เนื่องจาก HTML ที่ทำงานคู่กับ CSS จะมีแต่ส่วนเนื้อหาทำให้ง่ายต่อการจัดการและสามารถนำ CSS นั้นไปใช้กับ HTML อื่น ๆ ได้อีกด้วย จึงประหยัดเวลาในการพัฒนา (มีไฟล์เฉพาะในการตกแต่ง)
3. ใช้งานได้ดีกับเทคโนโลยีใหม่ ๆ เช่น การแสดงผลบนเบราว์เซอร์ใหม่ ๆ การแสดงผลบนรูปแบบอื่น ๆ อย่าง การสั่งพิมพ์ หรือบนมือถือ

➤ การเขียนโค้ด CSS แบบ Inline

แบบ Inline คือการเขียนโค้ด CSS ลงไปในแท็ก HTML ที่ต้องการให้เกิดการแสดงผลเลย และจะมีผลต่อแท็กนั้น ๆ เท่านั้น โดยมี Syntax ดังนี้

```
<tag style="property:value; property:value; . . .">
```

แอททริบิวต์ style ใช้ในการกำหนดค่าสไตล์หรือฟอร์ตต่าง ๆ ซึ่งจะอยู่ในรูปของ "property:value" หากมีมากกว่าหนึ่งชุดให้ใช้เครื่องหมาย Semicolon ";" คั่น ตัวอย่างเช่น

```
<h1 style="color:#0000FF">Where to Put CSS?</h1>?
```

ในตัวอย่างข้างต้นเรากำหนดสไตล์ให้กับแท็ก <h1> โดยให้ข้อความเป็นสีฟ้าด้วยสไตล์หรือฟอร์ต color

การเขียนแบบ Inline ไม่เป็นที่นิยมนัก เนื่องจากต้องเขียนโค้ด CSS ลงไปในแท็ก HTML ดังนั้นจึงไม่สามารถนำสไตล์ไปประยุกต์ใช้กับแท็กอื่น ๆ ได้ ลองนึกดูว่าถ้าต้องระบุค่าสีให้กับทุก ๆ แท็ก <h1> ในทุก ๆ หน้าเว็บคงจะลำบากน่าดูหากว่าต้องการเปลี่ยนสีในภายหลัง

➤ การเขียนโค้ด CSS แบบ Embedded

แบบที่ 2 แบบ Embedded เป็นการเขียนโค้ด CSS ลงไปในไฟล์ HTML เช่นกัน แต่จะรวมโค้ด CSS ทั้งหมดไว้ภายใต้แท็ก <head> โดยมี Syntax ดังนี้

การกำหนดสไตล์จะถูกกำหนดไว้ภายในแท็ก <style> ซึ่งอยู่ภายในแท็ก <head> อีกที การกำหนดสไตล์จะเริ่มต้นด้วยการระบุแท็กที่ต้องการ (Selector) จากนั้นจึงเป็นสไตล์พร็อพเพอร์ตี้ "property:value" ต่าง ๆ ภายในเครื่องหมายปีกกา แต่ละชุดจะถูกคั่นด้วยเครื่องหมาย Semicolon ";" ตัวอย่างเช่น

```
<head>
<title></title>
<style type="text/css">
  h1 {
    color: #0000FF;
    font-style: italic;
  }
  li {
    list-style-type: upper-roman;
  }
</style>
</head>
```

ในตัวอย่างข้างต้นเรากำหนดสีให้กับแท็ก <h1> เป็นสีฟ้า และใช้ตัวอักษรแบบตัวเอียง นอกจากนี้ยังกำหนดให้แต่ละหัวข้อของลิสต์ () แสดงเป็นเลขโรมันเรียงตามลำดับ เช่น i, ii, iii, iv เป็นต้น ตัวอย่างข้างต้นมีข้อสังเกตว่า คุณสมบัติต่าง ๆ ที่กำหนดจะมีผลต่อแท็ก <h1> และ ทุก ๆ แท็กที่ปรากฏในไฟล์ HTML ถ้ามี

การเขียนแบบ Embedded ก็ไม่เป็นที่นิยม เพราะต้องเขียนรวมอยู่ในไฟล์ HTML เช่นเดียวกับแบบ Inline หากจะนำไปใช้กับไฟล์ HTML อื่น ๆ ก็ต้องก๊อปปี้โค้ดเอา ดังนั้นการเปลี่ยนแปลงแก้ไขจึงค่อนข้างยุ่งยาก เพราะต้องตามไล่แก้ไขในทุก ๆ ไฟล์

➤ การเขียนโค้ด CSS แบบ External

แบบสุดท้ายแบบ External เป็นการเขียนแบบแยกโค้ด CSS ออกมายังไฟล์แยกต่างหาก ซึ่งชื่อก็บอกอยู่แล้วว่า External โดยมี Syntax ดังนี้

HTML:

```
<head>
  <title> </title>
  <link rel="stylesheet" type="text/css"
    href="fileName.css" />
</head>
```

CSS:

```
selector {
  property:value;
  property:value;
  ...
}
```

รูปแบบนี้จะแยกโค้ด CSS ไปไว้ในไฟล์แยก โดยมีนามสกุลเป็น ".css" ส่วนโครงสร้างการเขียนโค้ดจะเหมือนกับแบบ Embedded คือเริ่มต้นด้วยการระบุแท็ก (Selector) แล้วตามด้วยสไตล์หรือพเพอร์ตี "property:value" ต่าง ๆ ภายในเครื่องหมายปีกกา และโปรดสังเกตว่าในไฟล์แยกไม่ต้องมีแท็ก <style> นะครับ ให้มีแต่โค้ด CSS เท่านั้น จากนั้นในไฟล์ HTML ก็ให้ลิงค์มายังไฟล์ CSS ด้วยแท็ก <link> ตัวอย่างเช่น

HTML:

```
<head>
  <title></title>
  <link rel="stylesheet" type="text/css"
    href="External.css" />
</head>
```

CSS (External.css):

```
h1 {
  color: #0000FF;
  font-style: italic;
}
li {
  list-style-type: upper-roman;
}
```

ในตัวอย่างข้างต้นไฟล์ HTML และ CSS จะต้องอยู่ในโดเมนเดียวกัน (ไฟล์ CSS จะใช้ชื่ออะไรก็ได้) และถ้าไฟล์ HTML โดยอยากใช้สไตล์ที่กำหนดในไฟล์ External.css ก็สามารถลิงค์มาได้ โดยแค่เพิ่มแท็ก <link> เข้าไป ไม่เพียงเท่านี้ไฟล์ HTML ยังสามารถลิงค์ไปยังไฟล์ CSS ได้มากกว่าหนึ่งไฟล์ด้วย ทำให้เราสามารถแยกเป็นไฟล์ CSS ตามแต่ละจุดประสงค์ได้ เช่น Menu.css สำหรับจัดการกับเมนู, Layout.css จัดการกับเลย์เอาต์ เป็นต้น เว็บไซต์ใหญ่ ๆ ก็ใช้รูปแบบนี้ในการสร้างธีม (Theme) นะครับ แล้วก็ประยุกต์ใช้มันกับทุกหน้าในเว็บไซต์นั้น

การเขียนแบบ External เป็นรูปแบบที่นิยมที่สุดเพราะสามารถนำกลับมาใช้ใหม่ได้ ถ้าทุกหน้ามีสไตล์ที่ตรงกัน ก็ให้ลิงค์ไปที่ไฟล์ CSS เดียวกัน ดังนั้นหากต้องการเปลี่ยนแปลงรูปลักษณ์ใหม่ ก็เปลี่ยนมันทีเดียวเลยสะดวกดี

2.5.3 ภาษา PHP (Hypertext Preprocessor)

PHP คือภาษาสำหรับทำงานด้านฝั่งของเซิร์ฟเวอร์ (server-side scripting) ถูกออกแบบมาสำหรับการพัฒนาเว็บ แต่มันก็ยังสามารถใช้เขียนโปรแกรมเพื่อวัตถุประสงค์ทั่วไปได้ PHP ถูกสร้างโดย Rasmus Lerdorf ในปี 1994 โดยที่ PHP ในปัจจุบันได้ถูกพัฒนาโดยทีมพัฒนาของภาษา PHP ซึ่งคำว่า PHP นั้นย่อมาจาก Personal Home Page ซึ่งในปัจจุบันนั้นหมายถึง PHP: Hypertext Preprocessor

โค้ดของภาษา PHP นั้นสามารถฝังกับโค้ดของ HTML ได้ ซึ่งมันสามารถนำไปร่วมใช้ร่วมกับระบบเว็บที่หลายหลาย ระบบจัดการเนื้อหา (CMS) หรือเว็บเฟรมเวิร์ค การทำงานของภาษา PHP นั้นเป็นแบบ Interpreter ที่ถูกพัฒนาเป็นแบบโมดูลในเว็บเซิร์ฟเวอร์ หรือ Common Gateway Interface (CGI) โดยเซิร์ฟเวอร์จะทำการรวมโค้ดที่ผ่านการแปลผล และประมวลผลเป็นหน้าเว็บ และยังสามารถทำงานได้บน Command-line interface (CLI) และนอกจากนี้ภาษา PHP ยังถูกนำไปพัฒนาแอปพลิเคชันทางด้านกราฟิก

- Simple program

โปรแกรมแรกที่คุณจะได้เห็นในภาษา PHP จะเป็นโปรแกรมในการแสดงผลข้อความ "Hello World!" ออกทางหน้าจอ

```
<?php  
echo "Hello World!";  
?>
```

ในตัวอย่างเป็นโปรแกรมแสดงข้อความออกทางหน้าจอ คุณสามารถลองเปลี่ยนเป็นข้อความที่คุณต้องการได้เพื่อดูผลลัพธ์ของมัน

ในการเขียนโปรแกรมภาษา PHP โค้ดของโปรแกรมจะต้องอยู่ภายในบล็อกคำสั่ง <?php และ ?> เพื่อรันโปรแกรมสำหรับบน Windows เปิด Command line ขึ้นมาแล้วพิมพ์คำสั่ง php c:\project\hello.php ซึ่งเราได้ใช้คำสั่ง php ที่ได้จากการติดตั้งไปแล้วในบทก่อนหน้า และตามด้วยที่อยู่ของไฟล์ที่สคริปของ PHP อยู่

- การแสดงผลทางหน้าจอ

การแสดงผลในภาษา PHP นั้นมักจะใช้คำสั่ง echo และ print สำหรับจัดการและเชื่อมต่อกับ Output stream ซึ่งโดยปกติแล้วมักจะเป็นจอภาพหรือ Console มาดูตัวอย่างการแสดงผลในเบื้องต้น

```
<?php
echo "This text printed using echo.\n";
print "This text printed using print.\n";
?>
```

ในตัวอย่างเป็นการแสดงข้อความโดยการใช้นิพจน์ echo และ print ในภาษา PHP คุณจะได้เรียนเพิ่มเติมเกี่ยวกับการทำงานในภายหลัง

- PHP embedded in HTML

PHP นั้นเป็นภาษาสคริปต์ ดังนั้นมันจึงสามารถเขียนแทรกกับเอกสารรูปแบบอื่น ๆ ได้ซึ่งโดยส่วนมากแล้วมักจะนิยมเขียนกับ HTML เพื่อสร้างหน้าเว็บแบบไดนามิกส์

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Marcuscode.com</title>
</head>
<body>
<?php
echo "<h1>This text is generated by PHP</h1>";
?>
</body>
</html>
```

ในตัวอย่างเป็นไฟล์ในรูปแบบ HTML ซึ่งเราได้ทำการแทรกสคริปต์ของภาษา PHP ในระหว่าง <?php และ ?> อย่างไรก็ตามนามสกุลของไฟล์นั้นต้องเป็น .php เสมอ

นอกจากนี้แล้วภาษา PHP ยังสามารถใช้สร้างข้อมูลรูปแบบอื่นได้ โดยการกำหนด header content type ให้กับเอกสารนั้น เช่น JSON XML JavaScript หรือ CSS เป็นต้น กล่าวโดยสรุป PHP นั้นอำนวยความสะดวกในการสร้างเนื้อหาของของเว็บแบบไดนามิกส์ได้ ซึ่งจะทำให้การเขียนโปรแกรมยืดหยุ่นและง่ายมากขึ้น

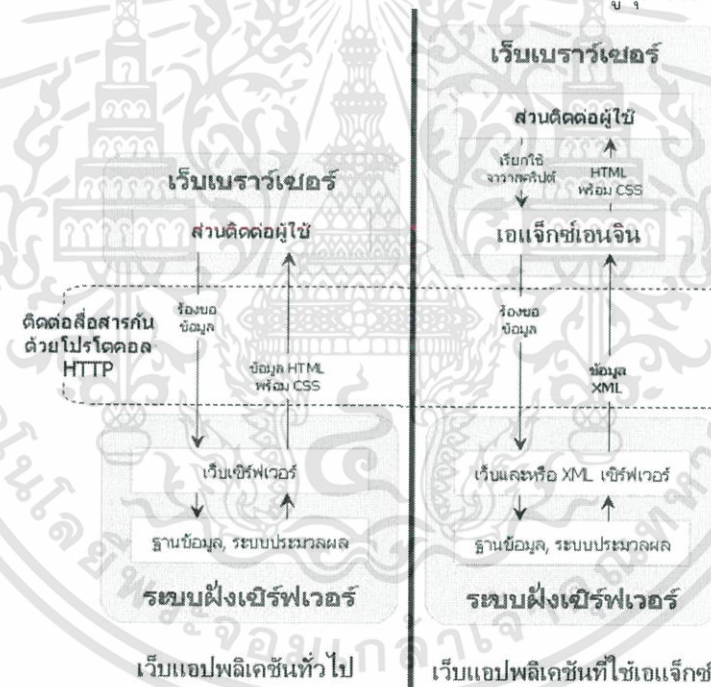
2.5.4 ภาษา JavaScript

JavaScript เป็นภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า “สคริปต์” (script) ซึ่งในการสร้างและพัฒนาเว็บ (ใช้ร่วมกับ HTML) เพื่อให้เว็บดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งก่อนที่จะเริ่มเรียนเรื่อง JavaScript นั้น จำเป็นต้องมีความรู้พื้นฐานเรื่อง HTML และ CSS เสียก่อน

ในการใช้งาน JavaScript นั้น จำเป็นต้องใส่โค้ดให้อยู่ระหว่างแท็ก <script> และ </script> โดยตัวคำสั่ง JavaScript นี้จะอยู่ในส่วนแท็ก <head> และ <body> ของเอกสาร HTML เช่น JavaScript ในแท็ก <body>

- **AJAX (Asynchronous JavaScript and XML)**

AJAX ย่อมาจาก Asynchronous JavaScript and XML มีความหมายว่า เป็นการทำงานร่วมกันของ JavaScript และ XML โดยจะเป็นการทำงานแบบที่ไม่ต้องรอคอย เมื่อ Browser ร้องขอข้อมูลไปยัง Server บราวเซอร์จะไปทำงานคำสั่งถัดไปทันที โดยที่ไม่ต้อง รอการตอบกลับจาก Server ก่อน ทำให้การตอบสนองต่อ User ดูรวดเร็วขึ้น นอกจากนี้ เรายังใช้ AJAX ในการร้องขอข้อมูลจาก Server โดยที่ไม่จำเป็นต้อง Reload หน้า เพื่อจัดการแสดงผลใหม่ และใช้ JavaScript เพื่อควบคุมการแสดงผลเพียงบางส่วนที่เปลี่ยนแปลง ทำให้การแสดงผลดูนุ่มนวล และรวดเร็วยิ่งขึ้น



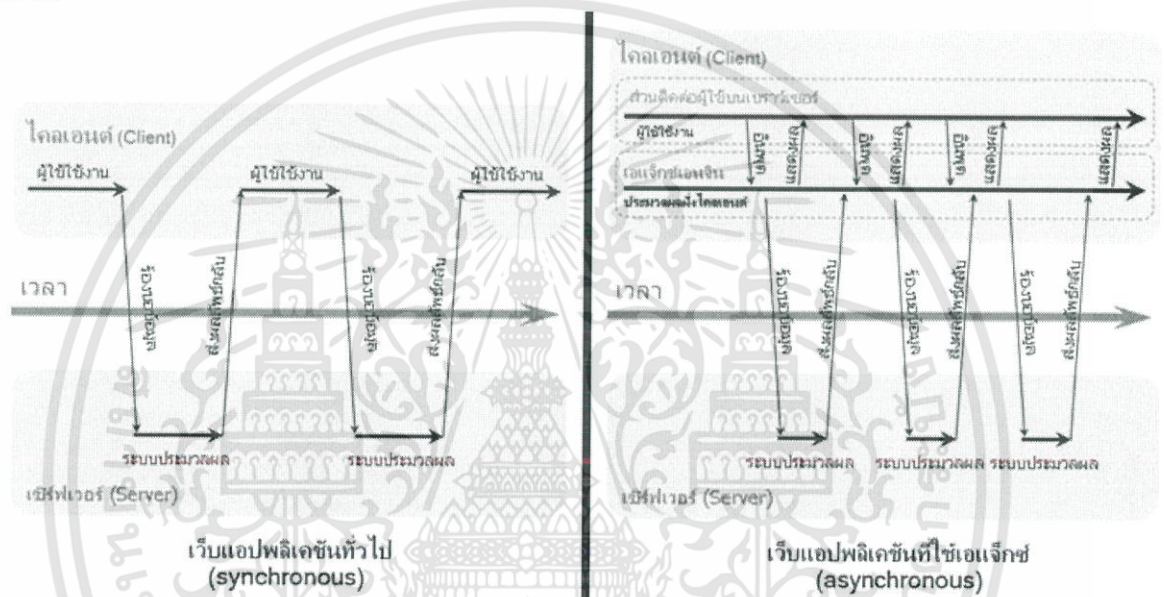
รูปที่ 2.19 การเปรียบเทียบระหว่างเว็บแอปพลิเคชันแบบดั้งเดิมกับแบบที่ใช้ AJAX

นอกจากนี้ เรายังสามารถใช้งาน AJAX ทำงานร่วมกับ JavaScript XML DHML CSS และ DOM ได้อีก เพื่อเสริม ประสิทธิภาพในการใช้งาน ให้เป็นระเบียบ และดูเรียบร้อยขึ้นได้ด้วย

บนหน้าเว็บทั่ว ๆ ไป การทำงานจะเริ่มต้นเมื่อ Browser ร้องขอข้อมูลหน้าเว็บไปยัง Server ซึ่ง Server จะทำการประมวลผลคำสั่งจนเสร็จ แล้วส่งเอกสารทั้งหน้ากลับมาที่ Browser เพื่อแสดงผลอีกที จากขั้นตอนต่าง ๆ จะเห็นได้ว่า เราจะต้องรอคอยให้ Server ประมวลผลเสร็จ รวมถึงรอคอยให้ Server ส่งข้อมูลกลับมาจนครบถ้วน จึงจะแสดงผลได้ ซึ่งถ้าเอกสารมีขนาดใหญ่ ก็จะทำให้การแสดงผลล่าช้าจน User เบื่อได้

AJAX ได้เข้ามาแก้ปัญหานี้ให้ โดยการที่เราจะแบ่งข้อมูลออกเป็นกลุ่มย่อย ๆ ตามความจำเป็น โดยที่เมื่อเราเรียกเพจครั้งแรก Browser อาจจะต้องรอข้อมูลเพียงบางส่วนก่อน เช่นอาจเรียกส่วนที่เป็น พื้นหลัง Logo หรือ ตารางหลักต่าง ๆ ออกมาก่อน ซึ่งจะทำให้ข้อมูลที่ Server ต้องตอบกลับมา มีขนาดน้อยลง ใช้เวลาโหลดและแสดงผลเร็วขึ้น ในขณะที่เดียวกัน AJAX จะทำหน้าที่ ร้องขอข้อมูลในส่วนที่เป็น ข้อมูล จาก Server พร้อม ๆ กับการจัดการแสดงผลในส่วนแรก (Asynchronous) ซึ่งเมื่อ Server ประมวลผลเสร็จเรียบร้อย Server จะส่งข้อมูลกลับมาที่ Browser อีกที และเราก็ใช้ JavaScript จัดการแสดงผลในส่วนที่เหลืออีกที

จะเห็นได้ว่า การแสดงผลจะดูรวดเร็วขึ้น ในสายตาของ User เพราะ Browser จะไม่ต้องรอคอยให้การส่งข้อมูลครบถ้วนก่อนถึงจะแสดงผลได้ ส่งผลดีต่อความรู้สึกของ User ที่มีต่อหน้าเว็บยิ่งขึ้น



รูปที่ 2.20 การเปรียบเทียบการสื่อสารระหว่างเว็บแอปพลิเคชันแบบดั้งเดิมกับแบบที่ใช้ AJAX

นอกจากนี้ จากการที่เราแบ่งการแสดงผลเป็นส่วน ๆ ทำให้การโหลดข้อมูลในหน้าต่อ ๆ ไป ไม่จำเป็นต้องโหลดข้อมูลทั้งหน้าขึ้นมาอีก เราจะโหลดเฉพาะในส่วนของการที่ต้องการเท่านั้นมา ทำให้การโหลดหน้าต่อ ๆ ไป รวดเร็วขึ้นมาก เพราะไม่ต้องโหลดใหม่ทั้งหน้า

ผลดีของการแยกการโหลดออกเป็นส่วน ๆ นอกจากที่กล่าวมาแล้วยังมีอีกอย่างหนึ่งคือ การแบ่งการแสดงผลออกเป็นส่วน ๆ ทำให้การเปลี่ยนแปลงข้อมูลดูนุ่มนวลขึ้น เพราะเราสามารถใช้ JavaScript ในการควบคุมการแสดงผลเป็นส่วน ๆ ได้ลดการโหลด และกระทบของหน้าจอ และยัง สามารถ แสดงผลในแบบ Real Time ได้โดยการโหลดข้อมูล เฉพาะที่มีการเปลี่ยนแปลงเท่านั้น ออกมาแสดง (Auto Refresh)

➤ Ajax JavaScript & Event

เนื่องจากการใช้งาน Ajax จะเป็นการเรียกใช้งาน Event ของ JavaScript ซึ่งพื้นฐานแล้ว จะมีไม่กี่ตัวครับ เช่น OnLoad(), OnClick(), OnDbClick() จึงอยากให้เราทำการศึกษาในส่วนของ Event ให้เข้าใจก่อน ถึงจะสามารถนำมาประยุกต์การใช้งานร่วมกับ Ajax ได้อย่างคล่องแคล่ว

➤ Ajax Display Element

เป็นคำสั่งที่ควบคุมซ่อนและแสดงผลของ Element ต่าง ๆ ภายในเว็บไซต์ และทำการซ่อน Element นั้นออกโดยไม่รักษา Layout เดิมไว้

`document.getElementById('myTable').style.display = '';` คือการแสดงผลออกมาทางหน้าเว็บ

`document.getElementById('myTable').style.display = 'none';` คือซ่อนการแสดงผล

➤ Ajax Disabled Element

เป็นคำสั่งที่ควบคุมการเปลี่ยนสถานะของ Input ทั้งหมด โดยสามารถ Disabled หรือ Enabled ในส่วนของ Element Input ที่ไว้รับค่า

`document.getElementById('txt1').disabled = true;` ปิดสถานะ

`document.getElementById('txt1').disabled = false;` เปิดสถานะปกติ

➤ Ajax readyState

จากตัวอย่างก่อนหน้านี้นี้จะเป็นการกำหนด Parameter(.open) เป็น false เช่น `HttpRequest.open('POST',url,false);` คือให้มีการทำงานทันทีทุกครั้งที่มีการ Request โดยไม่ต้องสนใจสถานะใด ๆ ทั้งสิ้น แต่ถ้ากำหนดเป็น Parameter(.open) เป็น true

เช่น `HttpRequest.open('POST',url,true);` ในคำสั่งของ JavaScript จะเป็นการควบคุมในส่วนของ Status การทำงาน ซึ่งจะทำการส่งสถานะกลับมาในแต่ละครั้งที่ทำงาน เพื่อให้ผู้ใช้ทำการเขียน Script เพื่อดักจับสถานะของการทำงานในขณะนั้น ๆ ซึ่งบางครั้งเราจะสังเกตเห็นว่า การทำงานค่อนข้างจะเร็วมาก แทบจะไม่มีข้อแตกต่างเลย

Status

0 การร้องขอ (request) มีปัญหา ไม่สามารถกำหนดค่าเริ่มต้นได้

1 การร้องขอ (request) ถูกกำหนดขึ้น

2 การร้องขอ (request) ถูกส่งไปแล้ว

3 การร้องขอ (request) กำลังประมวลผล

4 การร้องขอ (request) ให้ผลกลับมาเรียบร้อยแล้ว

➤ Ajax Realtime(PHP+MySQL & ASP+Access)

เป็นตัวอย่างการใช้ Ajax ในการดึงข้อมูลแบบ (Realtime) จากฐานข้อมูลมาแสดงผล มีตัวอย่างทั้ง PHP+MySQL และ ASP+Access ครบ และทั้งยังสามารถจัดเรียงการแสดงผลได้อีกด้วยครับ

หลักการ

ใช้ function `setTimeout()` ในการทำงานเป็น Loop โดยเรียกข้อมูลซ้ำ ๆ ตามระยะเวลาที่กำหนด

```

<script language="JavaScript">
function bodyOnload()
{
doCallAjax('CustomerID')
setTimeout("doLoop();",2000);
}
function doLoop()
{
bodyOnload();
}
</script>
โดย setTimeout("doLoop();",2000); 1,000 = 1 วินาที

```

- **jQuery Data Tables**

jQuery เป็น JavaScript Library ที่มีการรวบรวม function ของ JavaScript ต่าง ๆ ให้อยู่ในรูปแบบ Patterns Framework ที่สะดวกและง่ายต่อการใช้งาน มีความยืดหยุ่นรองรับต่อการใช้งาน Cross Browser คือไม่ว่าจะใช้งานบน Web Browser ใด ใน Library ของ jQuery จะมีการเลือกใช้ function ที่สามารถ เหมาะสมต่อการทำงานและแสดงผลใน Web Browser ที่กำลังรันอยู่ ซึ่งช่วยลดปัญหาการทำงานที่ผิดพลาดในฝั่งของ Client ได้ จากปัญหาก่อนนี้ นักโปรแกรมเมอร์ทั้งหลายในสมัยก่อน ๆ มักจะทดสอบโปรแกรมและพัฒนาบน IE (Internet Explorer ซึ่งเป็น Web Browser ที่คนใช้มากที่สุดเกือบ 95% เมื่อสมัย 5-6 ปี) แต่อย่างที่เรารู้คือ ตอนนี้ได้มีหลาย Web Browser ได้เกิดขึ้นมากมาย เช่น Chrome , Firefox หรือ Safari และบางคำสั่งของ JavaScript จะไม่ทำงานหรือไม่ support ใน Web Browser บางตัว

ด้วยเหตุผลนี้เองการใช้ jQuery มาเป็นทางเลือกก็สามารถช่วยแก้ปัญหานี้เป็นได้อย่างดี ทั้งยังสะดวกต่อการใช้งาน เพราะเป็น syntax ที่เข้าใจง่าย และเขียนได้ในรูปแบบที่สั้น ๆ รองรับการทำงานทั้งใน HTML รูปแบบเดิม หรือ CSS , element , DOM element , effect การจัดการ Event ต่าง ๆ หรือแม้กระทั่งการพัฒนา Ajax ด้วย jQuery ก็ทำได้ง่ายโดย Syntax เหล่านี้ยังคงทำงานอยู่ภายใต้คำสั่งของภาษา JavaScript แต่การเรียกใช้งาน Framework หรือ function ต่าง ๆ จะถูกกำหนดรูปแบบโดย Patterns ที่ได้ถูกออกแบบไว้ใน Library ของ jQuery



รูปที่ 2.21 Logo ของ jQuery

DataTables คือ plug-in jQuery JavaScript สำหรับ จัดการข้อมูล Html Table ซึ่ง การใช้งานนั้นง่ายมาก ๆ เพียงแค่ นำเข้า Script ก็สามารถใช้งานได้ทันที โดยจะมีการจัดรูปแบบข้อมูล ใน Table ของเรา ให้สามารถ ค้นหา (Search), เรียงลำดับ (Sort), แบ่งหน้า (Pagination) ให้อัตโนมัติ ซึ่งถือได้ว่าสุดยอดมาก ๆ และยังสามารถปรับแต่ง Options ต่าง ๆ ได้อีกด้วย

First name	Last name	Position	Office	Start date	Salary
Airi	Satou	Accountant	Tokyo	28th Nov 08	\$162,700
Angelica	Ramos	Chief Executive Officer (CEO)	London	9th Oct 09	\$1,200,000
Ashton	Cox	Junior Technical Author	San Francisco	12th Jan 09	\$86,000
Bradley	Greer	Software Engineer	London	13th Oct 12	\$132,000
Brenden	Wagner	Software Engineer	San Francisco	7th Jun 11	\$206,850
Brielle	Williamson	Integration Specialist	New York	2nd Dec 12	\$372,000
Bruno	Nash	Software Engineer	London	3rd May 11	\$163,500
Caesar	Vance	Pre-Sales Support	New York	12th Dec 11	\$106,450
Cara	Stevens	Sales Assistant	New York	6th Dec 11	\$145,600
Cedric	Kelly	Senior Javascript Developer	Edinburgh	29th Mar 12	\$433,060

Showing 1 to 10 of 57 entries

Previous 1 2 3 4 5 6 Next

รูปที่ 2.22 ตัวอย่างของ Data table

Javascript HTML CSS Comments (0)

The Javascript shown below is used to initialise the table shown in this example:

```

1 $(document).ready(function() {
2     $('#example').DataTable();
3 });

```

In addition to the above code, the following Javascript library files are loaded for use in this example:

<https://code.jquery.com/jquery-3.3.1.js>
<https://cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js>

รูปที่ 2.23 ตัวอย่างของโค้ดในเว็บ Data tables

2.6 โพรโทคอล HTTP

Hyper Text Transfer Protocol หรือ HTTP คือโพรโทคอลในระดับชั้นโปรแกรมประยุกต์ เพื่อการแจกจ่ายและการทำงานร่วมกันกับสารสนเทศของสื่อผสม ใช้สำหรับการรับข้อมูลที่เชื่อมโยงกับภายนอก ซึ่งนำไปสู่การจัดตั้งเว็บไซต์

เอชทีทีพีเป็นมาตรฐานในการร้องขอและการตอบรับระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่าย ซึ่งเครื่องลูกข่ายคือผู้ใช้ปลายทาง (End-user) และเครื่องแม่ข่ายคือเว็บ เครื่องลูกข่ายจะสร้างการร้องขอเอชทีทีพีผ่านทางเว็บเบราว์เซอร์ เว็บครอว์เลอร์ หรือเครื่องมืออื่น ๆ ที่จัดว่าเป็น ตัวแทนผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(User agent) ส่วนเครื่องแม่ข่ายที่ตอบรับ ซึ่งเก็บบันทึกหรือสร้าง ทรัพยากร (resource) อย่างเช่น ไฟล์เอชทีเอ็มแอลหรือรูปภาพ จะเรียกว่า เครื่องให้บริการต้นทาง (origin server) ในระหว่างตัวแทน ผู้ใช้กับเครื่องให้บริการต้นทางอาจมีสื่อกลางหลายชนิด อาทิพร็อกซี เกตเวย์ และทูนเนล เอชทีทีพี ไม่ได้จำกัดว่าจะต้องใช้ชุดเกณฑ์วิธีอินเทอร์เน็ต (TCP/IP) เท่านั้น แม้ว่าจะเป็นการใช้งานที่นิยมมากที่สุดบนอินเทอร์เน็ตก็ตาม โดยแท้จริงแล้วเอชทีทีพีสามารถ "นำไปใช้ได้บนโพรโทคอลอินเทอร์เน็ตอื่น ๆ หรือบนเครือข่ายอื่นก็ได้"

ปกติเครื่องลูกข่ายเอชทีทีพีจะเป็นผู้เริ่มสร้างการร้องขอ ก่อน โดยเปิดการเชื่อมต่อด้วยเกณฑ์วิธีควบคุมการขนส่งข้อมูล (TCP) ไปยังพอร์ตเฉพาะของเครื่องแม่ข่าย (พอร์ต 80 เป็นค่าปริยาย) เครื่องแม่ข่ายเอชทีทีพีที่เปิดรอรับอยู่ที่พอร์ตนั้น จะเปิดรอให้เครื่องลูกข่ายส่งข้อความร้องขอเข้ามา เมื่อได้รับการร้องขอแล้ว เครื่องแม่ข่ายจะตอบรับด้วยข้อความสถานะอันหนึ่ง ตัวอย่างเช่น "HTTP/1.1 200 OK" ตามด้วยเนื้อหาของมันเองส่งไปด้วย เนื้อหานี้ อาจเป็นแฟ้มข้อมูลที่ร้องขอ ข้อความแสดงข้อผิดพลาด หรือข้อมูลอย่างอื่น เป็นต้น

ทรัพยากรที่ถูกเข้าถึงด้วยเอชทีทีพีจะถูกระบุโดยใช้ตัวระบุแหล่งทรัพยากรสากล (URI) (หรือเจาะจงลงไปก็คือ ตัวชี้แหล่งในอินเทอร์เน็ต (URL)) โดยใช้ http: หรือ https: เป็นแผนของตัวระบุ (URI scheme)

คำสั่งร้องขอ

ตัวอย่างข้อความร้องขอเอชทีทีพีที่สร้างในเทลเน็ต เน้นสีส่วนหัวและส่วนเนื้อหาของทั้งข้อความร้องขอและข้อความตอบรับ

เอชทีทีพีได้กำหนดคำสั่งร้องขอไว้แปดคำสั่ง (หรือเรียกว่าวิธีการร้องขอ บางครั้งอาจเรียกว่าเป็น "กริยา") แสดงการกระทำที่ต้องการ เพื่อที่จะดำเนินการกับทรัพยากรที่ถูกระบุ สิ่งที่ทรัพยากรนั้น นำเสนอ ไม่ว่าจะเป็นข้อมูลที่มีอยู่ก่อนหรือสร้างขึ้นแบบพลวัตก็ตาม จะขึ้นอยู่กับการนำไปใช้ของเครื่องแม่ข่าย ซึ่งบ่อยครั้งทรัพยากรมักจะสอดคล้องกับไฟล์ หรือผลลัพธ์ส่งออกจากโปรแกรมข้างเคียงในเครื่องแม่ข่ายนั้น เครื่องให้บริการเอชทีทีพีจะต้องสามารถใช้คำสั่ง GET และ HEAD ได้เป็นอย่างน้อย

- HEAD

ร้องขอการตอบรับจากทรัพยากรที่ระบุ คล้ายกับ GET แต่จะไม่มีส่วนเนื้อหาที่ร้องขอกลับมา คำสั่งนี้ใช้ประโยชน์ในการตรวจสอบข้อมูลส่วนหัวของการตอบรับ โดยไม่จำเป็นต้องส่งเนื้อหาเต็มมาทั้งหมด

- GET

ร้องขอการนำเสนอจากทรัพยากรที่ระบุ คำสั่งนี้ไม่ควรใช้กับการดำเนินการที่อาจทำให้เกิดผลข้างเคียง เช่นการจัดการในเว็บแอปพลิเคชัน เหตุผลหนึ่งคือคำสั่ง GET มักจะถูกใช้อย่างไม่มีกฎเกณฑ์โดยอินเทอร์เน็ตบอตและเว็บครอว์เลอร์ ซึ่งไม่ควรพิจารณาให้การร้องขอของบอตและครอว์เลอร์ส่งผลกระทบต่อทรัพยากรในเว็บ (ดูเพิ่มที่หัวข้อ คำสั่งที่ปลอดภัย)

- POST

ส่งข้อมูลไปยังทรัพยากรที่ระบุเพื่อให้นำไปประมวลผล โดยเฉพาะข้อมูลที่ส่งมาจากฟอร์มเอชทีเอ็มแอล ข้อมูลที่ส่งจะถูกบรรจุอยู่ในเนื้อหาของการร้องขอด้วย สิ่งนี้อาจทำให้เกิดการสร้างทรัพยากรใหม่ หรือการปรับปรุงทรัพยากรที่มีอยู่ หรือทั้งสองกรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ 26 ขาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PUT
อัปเดตการนำเสนอของทรัพยากรที่ระบุ
- DELETE
ลบทรัพยากรที่ระบุ
- TRACE
ส่งข้อมูลร้องขอกลับมา เครื่องลูกข่ายจะเห็นว่าข้อมูลอะไรบ้างที่สื่อกลางเพิ่มหรือเปลี่ยนแปลงข้อความร้องขอก่อนไปถึงทรัพยากรปลายทาง
- OPTIONS
คืนค่าเป็นรายชื่อคำสั่งเฮชทีทีพีที่เครื่องแม่ข่ายนั้นรองรับสำหรับทรัพยากรที่ระบุ สิ่งนี้สามารถใช้ตรวจสอบฟังก์ชันการทำงานของเว็บเซิร์ฟเวอร์ได้โดยใส่ "*" แทนที่การระบุทรัพยากร
- CONNECT
แปลงการเชื่อมต่อของการร้องขอไปเป็นทูนเนล TCP/IP แบบโปร่งใส มักใช้สำหรับแปลงการเชื่อมต่อที่เข้ารหัสแบบ SSL ให้เดินทางผ่านพร็อกซีที่ไม่มีการเข้ารหัสได้ง่ายขึ้น



บทที่ 3

การแสดงผลข้อมูลระยะไกลที่นำเสนอ

3.1 กล่าวนำ

ขั้นตอนการออกแบบเว็บสำหรับแสดงผลค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าจากการวัดระยะไกลนั้น จะแบ่งออกเป็นสี่ขั้นตอน คือ การออกแบบและสร้างเว็บ, การเขียนโปรแกรมส่วนแสดงผลแบบเรียลไทม์ และสร้างฐานข้อมูลจำลองเพื่อทดสอบระบบ, การเชื่อมต่อกับฐานข้อมูลของบอร์ด ASUS Tinker โดยดึงข้อมูลจาก Web service และการแสดงค่าข้อมูลย้อนหลังโดยสามารถเลือกวันที่ได้

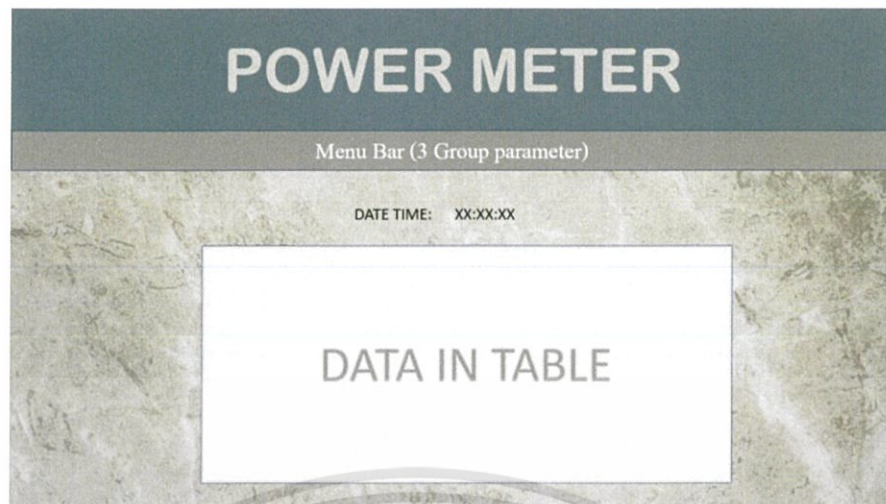
3.2 การออกแบบและสร้างเว็บสำหรับการแสดงผลที่นำเสนอ

ในเว็บหนึ่งเว็บ จะประกอบไปด้วยหน้าเว็บที่มีได้ตั้งแต่หนึ่งเพจเป็นต้นไป โดยในเว็บที่มีหน้าเว็บตั้งแต่สองเพจขึ้นไปนั้นจะมีเมนูบาร์ในการนำทางเพื่อเปลี่ยนจากหน้าเว็บหนึ่งไปยังอีกหน้าเว็บหนึ่ง ขั้นตอนต่อไปนี้จะกล่าวถึงการสร้างเว็บที่มีหน้าเว็บทั้งหมด 4 เพจ และมีเมนูบาร์เพื่อความสะดวกในการใช้งานของ User

3.2.1 การออกแบบเว็บ

ในการออกแบบเว็บสำหรับแสดงผลนั้น จำเป็นต้องทราบว่าบนหน้าเว็บแต่ละเพจจะแสดงค่าพารามิเตอร์ใดบ้าง เพื่อนำมาจัดเตรียมโครงร่างของหน้าเว็บก่อนการทำงาน ซึ่งจากข้อมูลที่ได้รับพบว่าค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าที่ต้องการนำมาแสดงนั้นถูกแบ่งออกเป็น 3 กลุ่ม ได้แก่ กลุ่มพลังงานรวมของระบบไฟฟ้าสามเฟส, กลุ่มค่าพารามิเตอร์พื้นฐาน และกลุ่มค่าพลังงานในแต่ละเฟส

และเว็บจะต้องแสดงผลข้อมูลออกมาในรูปแบบของเรียลไทม์และแบบข้อมูลย้อนหลัง โดยจะแสดงผลข้อมูลทั้งหมดในรูปแบบตาราง จึงแบ่งให้เว็บมีทั้งหมด 4 เพจ โดย 3 เพจแรกเป็นการแสดงผลข้อมูลแบบเรียลไทม์ตามกลุ่มพารามิเตอร์ และหน้าสุดท้ายเป็นการแสดงผลข้อมูลแบบย้อนหลัง โดยผู้ใช้งานสามารถกรอกวันที่เพื่อเลือกแสดงผลข้อมูลในวันที่ที่ต้องการได้ จากนั้นใช้โปรแกรม PowerPoint ออกแบบโครงร่างของเว็บขึ้นมา



รูปที่ 3.1 การออกแบบหน้าเว็บด้วยโปรแกรม PowerPoint

เมื่อได้ภาพร่างของหน้าตาเว็บแล้ว จึงเริ่มทำการเขียนโปรแกรมเพื่อสร้างเว็บออกมาตามโครงร่างที่ได้ออกแบบไว้ โดยใช้โปรแกรม Visual Studio Code ในการเขียนโค้ด และมีโปรแกรม Xampp เป็นเซิร์ฟเวอร์จำลองเพื่อสามารถทดสอบเว็บที่สร้างได้ในทันที

3.2.2 การเขียนโปรแกรมเพื่อสร้างเว็บ

การสร้างเว็บขึ้นมานั้นจะอิงกับโครงร่างที่เราได้ร่างไว้ในขั้นต้น โดยภาพพื้นหลังของเว็บที่นำมาใช้นั้นได้ค้นหาจากเว็บไซต์ <https://stocksnap.io/> ซึ่งเป็นภาพประเภท free license ที่เจ้าของภาพได้อนุญาตให้สามารถนำไปใช้ได้อย่างอิสระโดยไม่ติดลิขสิทธิ์

ในการเขียนเว็บนั้น จะมีการนำภาษาโปรแกรมมาใช้ร่วมกันทั้งหมด 3 ภาษา อันได้แก่ HTML ซึ่งเป็นการเขียนคอนเทนต์ต่าง ๆ ของหน้าเว็บ, CSS ซึ่งเป็นภาษาสำหรับตกแต่งหน้าเว็บให้สวยงาม และ JavaScript ซึ่งนำมาใช้ในการทำให้เว็บสามารถตอบสนองผู้ใช้งานได้มากขึ้น

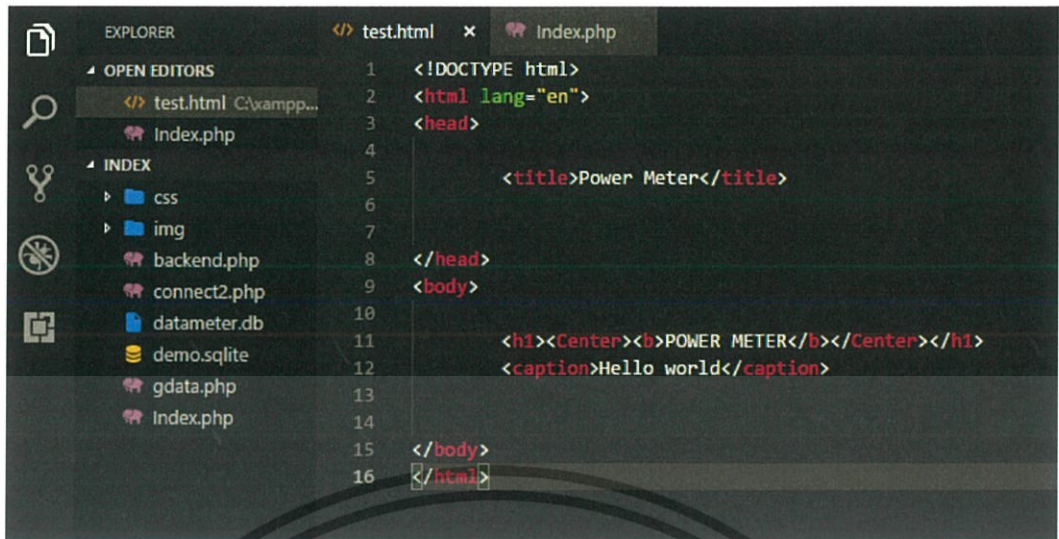
3.3.2.1 การสร้างหน้าเว็บ

➤ ส่วนของ HTML

หลักการในการเขียน HTML เพื่อสร้างเว็บนั้น จำเป็นต้องเข้าใจโครงสร้างของภาษา HTML ด้วย ซึ่งโครงสร้างของภาษา HTML จะประกอบด้วย

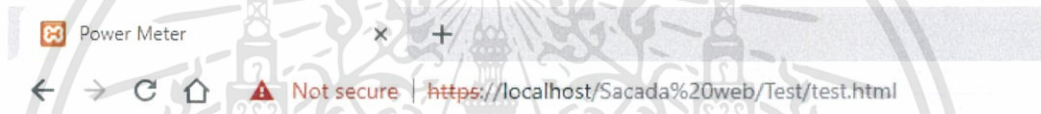
ส่วน Head – แสดงผลอยู่บนแท็บบาร์ของเว็บเบราว์เซอร์

ส่วน Body – เนื้อหาภายในหน้าเว็บนั้น ๆ



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4
5     <title>Power Meter</title>
6
7
8 </head>
9 <body>
10
11     <h1><Center><b>POWER METER</b></Center></h1>
12     <caption>Hello world</caption>
13
14
15 </body>
16 </html>
```

รูปที่ 3.2 ตัวอย่างการเขียนโค้ด HTML เบื้องต้น



รูปที่ 3.3 การแสดงผลของโค้ด HTML ส่วนของ Head



รูปที่ 3.4 การแสดงผลของโค้ด HTML ส่วนของ Body

➤ ส่วนของ CSS

ในส่วนของ HTML นั้น จะเห็นได้ว่าหน้าเว็บปรากฏเป็นตัวอักษรขาวดำธรรมดา จึงได้นำภาษา CSS มาใช้สำหรับการตกแต่งเว็บให้เป็นไปตามโครงสร้างที่ได้ออกแบบเอาไว้

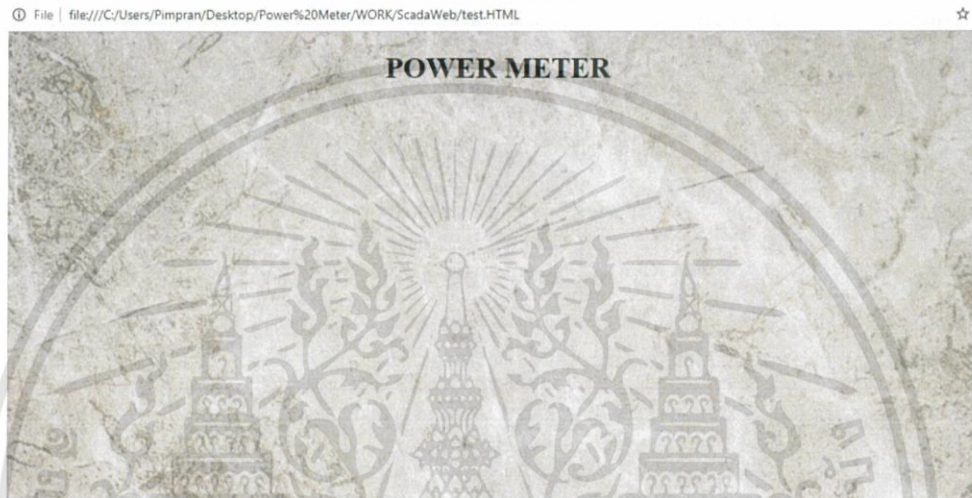
การใช้งาน CSS ให้ง่ายคือการเขียน CSS แบบ External คือแยกไฟล์โค้ด “.css” ออกมาอีกไฟล์ และระบุส่วนที่เราต้องการตกแต่งด้วย selector แบบแบ่ง class

1. การใส่ภาพพื้นหลัง

จากที่ได้เตรียมรูปภาพพื้นหลังแบบ free license เอาไว้ จะทำการดึงรูปภาพนั้นมาใช้โดยเขียนโค้ดระบุตำแหน่งที่เก็บบันทึกรูปภาพเพื่อนำมาใช้เป็นภาพพื้นหลัง ในที่นี้ตำแหน่งของรูปภาพคือ "img/1.jpg"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

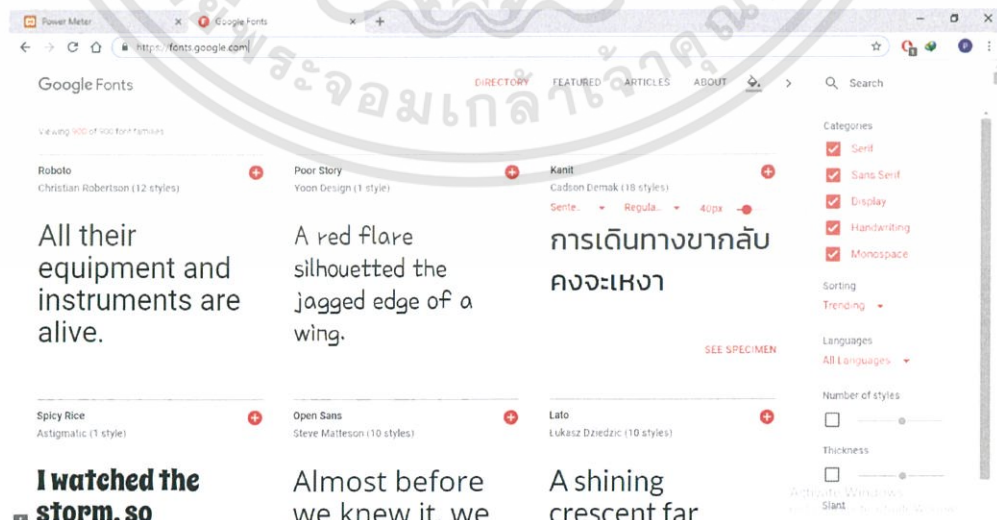
```
body {
    background-image: url("img/1.jpg");
    background-color: #cccccc;
    background-size: 100%;
}
```



รูปที่ 3.5 การแสดงผลเมื่อใส่ภาพพื้นหลังแล้ว

2. การเปลี่ยนฟอนต์ตัวอักษร

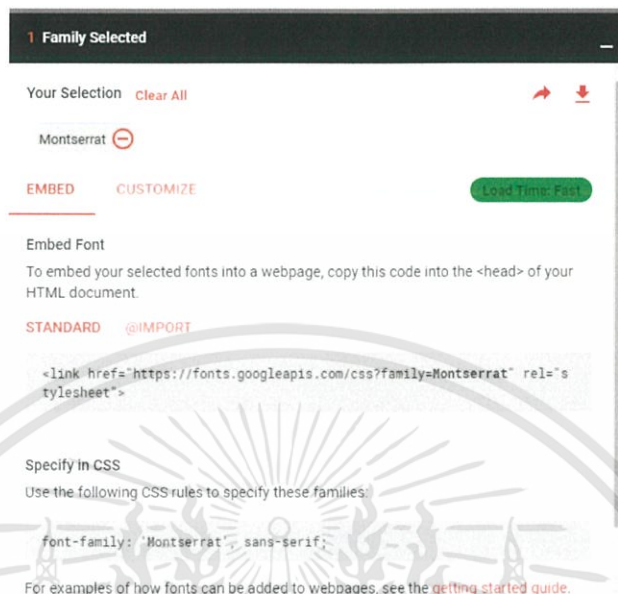
รูปแบบของฟอนต์ดั้งเดิมนั้นไม่สวยงามสำหรับเว็บที่ใช้ในการติดต่อกับผู้ใช้โดยตรง จึงต้องทำการเปลี่ยนฟอนต์เพื่อตอบสนองต่อความสวยงามและพึงพอใจของผู้ใช้งาน โดยสามารถเลือกฟอนต์รูปแบบต่าง ๆ ได้จากเว็บไซต์ <https://fonts.google.com/>



รูปที่ 3.6 เว็บไซต์ <https://fonts.google.com/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำฟอนต์จาก <https://fonts.google.com/> มาใช้งานนั้นทำได้ด้วยการคลิกรูป + ที่บริเวณมุมขวาบนของฟอนต์รูปแบบที่ต้องการ หลังจากนั้นจะปรากฏหน้าต่างรูปที่ 3.7 ขึ้นมา



รูปที่ 3.7 การใช้งานฟอนต์จากเว็บไซต์ <https://fonts.google.com/>

ให้คัดลอกโค้ดในช่องแรก Standard @Import ไปใส่ไว้ในส่วน HTML และคัดลอกโค้ดช่อง Specify in CSS ไปใส่ไว้ในไฟล์ css

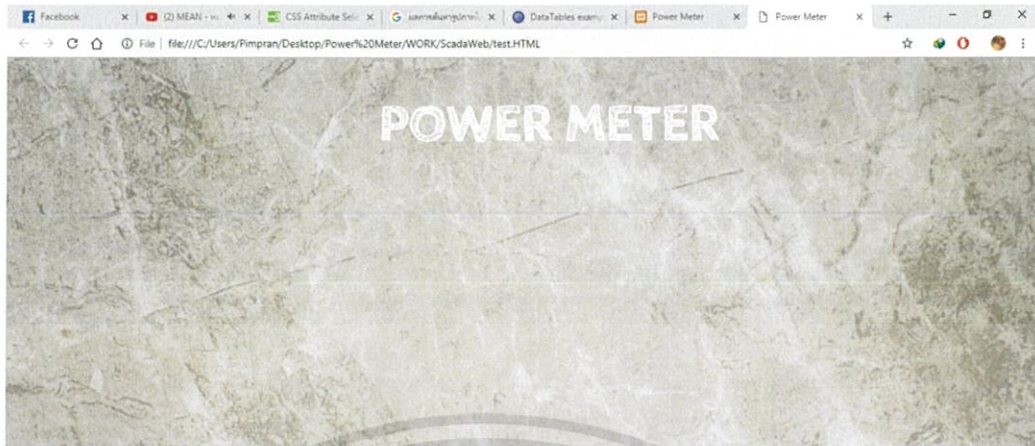
```
<code>test.html • css.css • Index.php • style1.css
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4
5 <title>Power Meter</title>
6
7
8 <link href="https://fonts.googleapis.com/css?family=Montserrat|Prompt" rel="stylesheet">
9
10 <link rel="stylesheet" href="css.css" > //การนำไฟล์CSSเข้ามาใช้
11 </head>
12</code>
```

รูปที่ 3.8 การเปลี่ยนรูปแบบฟอนต์ในส่วน HTML

```
css.css - Test - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
EXPLORER test.html • css.css • Index.php • style1.css
OPEN EDITORS 2 UNSAVED
• </> test.html
• css.css 2
Index.php C:\xamp...
style1.css C:\xamp...
1 body {
2 font-size: 20px; //การเปลี่ยนขนาดตัวอักษร
3 font-family: 'Montserrat', sans-serif;
4 }
```

รูปที่ 3.9 การเปลี่ยนรูปแบบฟอนต์ในส่วน CSS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 การแสดงผลเมื่อเปลี่ยนรูปแบบฟอนต์บนหน้าเว็บ

การใช้ฟอนต์สองรูปแบบทำได้โดยการใช้การแบ่ง class ของ CSS มาช่วย เพื่อบอกว่าบริเวณ class นี้ให้แสดงผลด้วยฟอนต์รูปแบบนี้ ซึ่งหากฟอนต์นั้นอยู่ใน family font เดียวกัน สามารถย่อได้ด้วยการใช้โค้ดเพียงครั้งเดียว ดังเช่น

font-family: 'Montserrat', 'Prompt', sans-serif;

```
<div class="Head">
  <h1><Center><b>POWER METER</b></Center></h1>
</div>

<div class="info1"><center>
  <caption>ค่าพารามิเตอร์ และปริมาณพลังงานไฟฟ้า</caption>
<br>
</div>
```

รูปที่ 3.11 การใช้ class ใน HTML เพื่อตกแต่งแต่ละส่วน

```
1  body {
2    font-size: 20px;
3    font-family: 'Montserrat', 'Prompt', sans-serif;
4  }
5  .Head {
6    font-family: 'Cabin Sketch', cursive;
7    font-size: 30px;
8  }
```

รูปที่ 3.12 การใช้ class ใน CSS เพื่อตกแต่งแต่ละส่วน

หมายเหตุ หากต้องการแสดงค่าภาษาไทยในบนเว็บ จะต้องทำการเขียนโค้ดเข้าไปตั้งค่า เพื่อให้โปรแกรมเรียกใช้งานภาษาไทยขึ้น โดยเขียนโค้ดลงไปในส่วน of <head>....</head> ดังนี้

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
```

3.3.2.2 การสร้างเมนูบาร์

ในส่วนของเมนูบาร์นั้นจะใช้ function Onclick ของ JavaScript สร้างเงื่อนไขในการกดปุ่มในแต่ละปุ่ม เพื่อช่วยให้คลิกปุ่มเพื่อเปลี่ยนหน้าเว็บโดยไม่ต้องโหลดทั้งเว็บใหม่ แต่โหลดเฉพาะข้อมูลที่ต้องการดึงเข้ามาแสดง

โดยเงื่อนไขในการกดปุ่มคือ เมื่อเมนูบาร์มีให้เลือกอยู่ทั้งหมด 4 เมนู สร้างปุ่มขึ้นมา 4 ปุ่ม ตั้งเงื่อนไขให้ทุกครั้งที่เปิดเว็บครั้งแรก เว็บจะแสดงเนื้อหาของหน้าเว็บแรกบนเมนูบาร์โดยอัตโนมัติ และเมื่อมีการกดปุ่มอื่นในเมนูบาร์ เงื่อนไขใน function Onclick จะถูกตรวจสอบ หากเงื่อนไขตรงกับหน้าเว็บใด ข้อมูลในหน้าเว็บนั้นจะถูกเรียกขึ้นมาแสดงบนเว็บ

โดยจะใช้โค้ด <button class="... ชื่อ class ที่เรากำหนด....." onclick="openCity(event,'split')"...ชื่อเมนู.....</button> ในการสร้างปุ่มบนเมนูบาร์ จากนั้นตกแต่งให้สวยงามด้วย CSS

- โค้ดส่วน HTML

```
<body>
  <div class="w3-container">
    <div class="w3-bar w3-black">
      <button class="w3-bar-item w3-button tablink w3-red"
onclick="openCity(event,'total')>Total</button>
      <button class="w3-bar-item w3-button tablink"
onclick="openCity(event,'split')>ค่าปริมาณพลังงานไฟฟ้า</button>
      <button class="w3-bar-item w3-button tablink"
onclick="openCity(event,'other')>Parameter</button>
      <button class="w3-bar-item w3-button tablink"
onclick="openCity(event,'print')>Historian</button>
    </div>
  </div>
<br>
<div id="total" class="w3-container w3-border city">
  // ส่วนข้อมูลของ เพจ Total
</div>
<div id="split" class="w3-container w3-border city" style="display:none">
  // ส่วนข้อมูลของ เพจ ค่าปริมาณพลังงานไฟฟ้า
</div>
<div id="other" class="w3-container w3-border city" style="display:none">
  // ส่วนข้อมูลของ เพจ Historian
</div>
<div id="print" class="w3-container w3-border city" style="display:none">
  // ส่วนข้อมูลของ เพจ Parameter
```

```
</div>

<script>
function openCity(evt, cityName) {
  var i, x, tablinks;
  x = document.getElementsByClassName("city");
  for (i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
  tablinks = document.getElementsByClassName("tablink");
  for (i = 0; i < x.length; i++) {
    tablinks[i].className = tablinks[i].className.replace(" w3-red", "");
  }
  document.getElementById(cityName).style.display = "block";
  evt.currentTarget.className += " w3-red";
}
</script>
</body>
```

- โค้ดส่วน CSS สามารถตกแต่งสี ขนาดตัวอักษร ขนาด padding, margin ได้ตามต้องการ



รูปที่ 3.13 ตัวอย่างการแสดงผลเมื่อสร้างเมนูบาร์เสร็จสิ้น

3.3 การเขียนโปรแกรมส่วนแสดงผลและสร้างฐานข้อมูลจำลองเพื่อทดสอบระบบ

3.3.1 สร้างฐานข้อมูลจำลองและเชื่อมต่อฐานข้อมูล

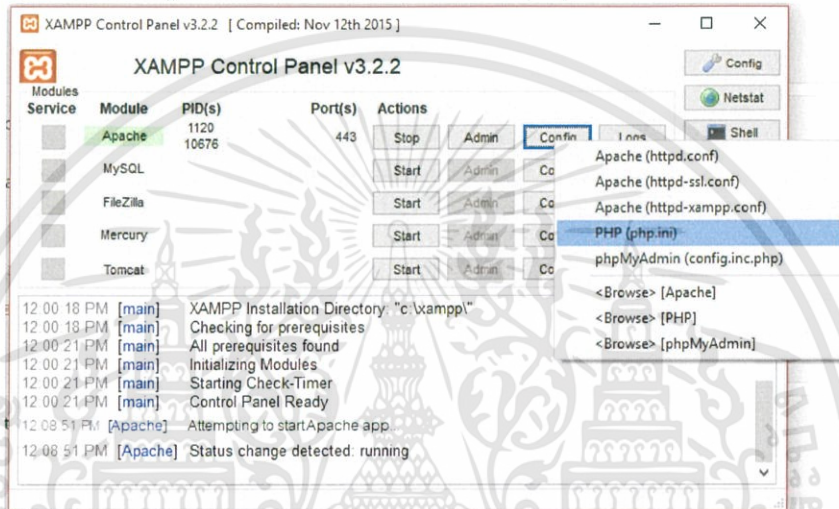
ในส่วนนี้คือการทดลองสร้างฐานข้อมูลจำลองขึ้นมาเพื่อใช้ในการทดสอบการเรียกค่าพารามิเตอร์จากฐานข้อมูลขึ้นมาแสดงค่า เพราะยังไม่สามารถดึงค่าจากฐานข้อมูลของบอร์ด ASUS

Tinker ได้โดยตรง หากไม่ทดลองสร้างฐานข้อมูลจำลองขึ้นมาก่อน จะไม่สามารถตรวจสอบว่าเว็บสามารถแสดงค่าแบบเรียลไทม์ได้ตามต้องการหรือไม่

โดยสามารถสร้างฐานข้อมูลขึ้นมาจากโปรแกรม SQLite3 และสร้างหน้าโค้ด Back-end จำลองขึ้นมาเพื่อใช้ในการเขียนโค้ดเรียกใช้งานฐานข้อมูล ซึ่งการทดลองในส่วนนี้จะสร้างทั้ง Front-end และ Back-end สำหรับทดสอบใหม่ขึ้นมา เพื่อง่ายต่อการแก้ไขและจัดการหากเกิดข้อผิดพลาดในการเขียนโค้ดต่าง ๆ

ในโปรแกรม Xampp นั้นจะต้องเปิดใช้งานส่วนขยายของ SQLite ใน Xampp ก่อนจึงจะสามารถเชื่อมต่อเข้ากับฐานข้อมูลจำลองที่สร้างขึ้นมาได้

การเข้าไป Configuration เข้าได้โดยการเปิดไฟล์ php.ini



รูปที่ 3.14 การ Configuration Apache เพื่อใช้งาน SQLite

ให้ลบตัว Semicolon (;) ออกจาก sqlite extension เพื่อเป็นการเปิดใช้งานส่วนขยายจากนั้นกดบันทึกและ Restart apache อีกครั้ง

เมื่อเปิดใช้งานฐานข้อมูลในเซิร์ฟเวอร์จำลองได้เรียบร้อยแล้ว ขั้นตอนต่อไปคือการเชื่อมต่อฐานข้อมูลเพื่อให้สามารถเรียกออกมาใช้งานได้ โดยโค้ดในส่วนนี้จะใช้เป็นภาษา PHP ในการเขียนโปรแกรม

ไฟล์โปรแกรมทั้ง Front-end และ Back-end จำลองจะต้องเป็นนามสกุล .php เพื่อให้สามารถ Run ภาษา PHP ได้ โดยภาษา PHP นั้นจะถูกเขียนในรูปแบบ <?php ?>

➤ การเชื่อมต่อฐานข้อมูล

สร้างไฟล์ Connect2.php แยกออกมาอีกไฟล์เพื่อง่ายต่อการเขียนโปรแกรม ซึ่งโค้ดที่ใช้ในการเรียกฐานข้อมูลมาใช้จะเป็นดังนี้

```

1 <?php
2 class MyDB extends SQLite3 {
3     function __construct() {
4         $this->open('demo.sqlite'); //ชื่อไฟล์ฐานข้อมูลที่สร้างเอาไว้
5     }
6 }
7 $db = new MyDB();
8 if (!$db) {
9     echo $db->lastErrorMsg();
10    exit();
11 }
12

```

รูปที่ 3.15 โค้ดเชื่อมต่อกับฐานข้อมูลของ SQLite

- การเรียกไฟล์ PHP ขึ้นมาใช้
เมื่อสร้างไฟล์ PHP สำหรับเชื่อมต่อกับฐานข้อมูลขึ้นมาแล้ว ในส่วน Back-end จำเป็นต้องเรียกไฟล์นั้นขึ้นมาใช้ โดยมีโค้ดเรียกใช้งานไฟล์ PHP ดังนี้

```

<?php
include('connect2.php');
?>

```

- การ Select ค่าจากฐานข้อมูล
การดึงค่าข้อมูลออกมาจากฐานข้อมูลจะใช้ภาษา SQL ในการเลือก Select ตารางในฐานข้อมูลนั้น

```

<?php
$sql =<<<EOF
SELECT * from PowerMeter;
EOF;

$ret = $db->query($sql);
while($row = $ret->fetchArray(SQLITE3_ASSOC) ) {
    echo ".$row[";}]

$db->close();
?>

```

.\$row['.....'] เป็นการชี้ไปที่ชื่อของแถวข้อมูล เมื่อทราบรูปแบบของโค้ดที่ใช้ในการเรียกข้อมูลจากฐานข้อมูลมาแสดงแล้ว จึงทำการสร้างตารางด้วยภาษา PHP เพื่อใช้ในการแสดงข้อมูลจากฐานข้อมูล

ตัวอย่างการเรียกข้อมูลจากฐานข้อมูลจำลองขึ้นแสดงบนหน้าเว็บในรูปแบบตาราง ทำได้โดยใช้โค้ด HTML <table>....</table> สร้างตารางขึ้นมา แล้วนำ \$.row['.....'] ไปใส่เป็นค่าในช่องตารางนั้น

```
<?php
    include('connect2.php');
?>

<?php $sql =<<<EOF
    SELECT * from PowerMeter;
    EOF;
echo "<table width='50%' cellspacing='1' align='center' bgcolor='#CCCCCC'>";
echo "<tr aling='center'>";
echo "<td align='center' bgcolor='#FFFFCC' width='2%'>&nbsp;&nbsp;&nbsp;</td>";
    <th align='center' bgcolor='#FFDAB9' width='20%'>Voltage (Volts)</th>
    <th align='center' bgcolor='#AFEEEE' width='20%'>Current (Amps)</th>
    <th align='center' bgcolor='#FFCCCC' width='20%'>Power (Watts)</th>";

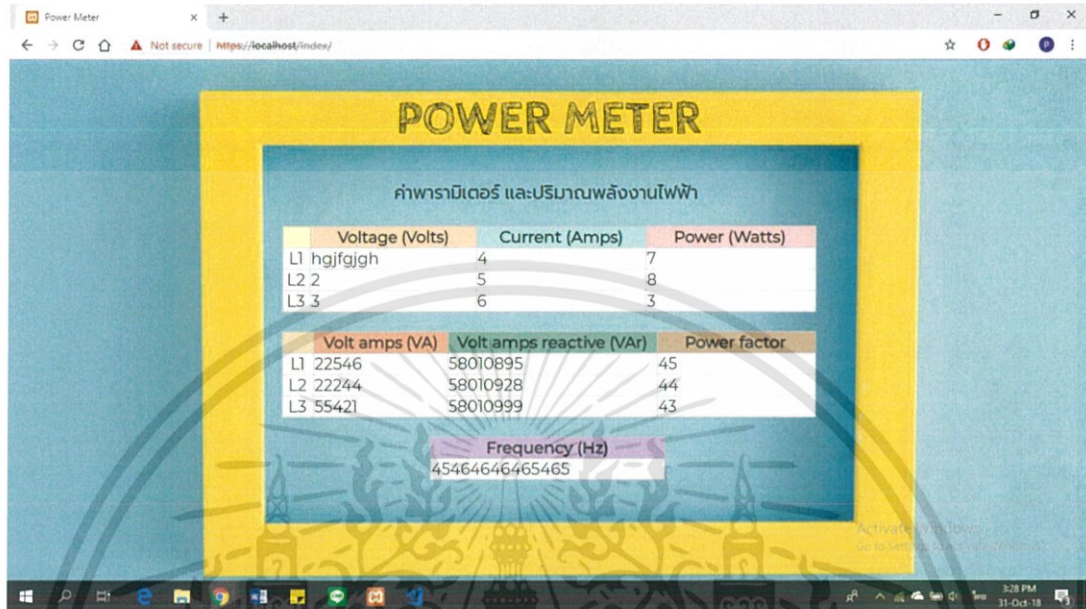
$ret = $db->query($sql);
while($row = $ret->fetchArray(SQLITE3_ASSOC) ) {
    echo "<tr bgcolor='#FFFFFF'>";
    echo "<td align='center'>" . "L1";
    echo "<td>" . $row['Voltage1'] . "</td> ";
    echo "<td>" . $row['Current1'] . "</td> ";
    echo "<td>" . $row['Power1'] . "</td> ";
    echo "</tr>";
echo "</table>";
$ret = $db->query($sql);
while($row = $ret->fetchArray(SQLITE3_ASSOC) ) {
    echo "<tr bgcolor='#FFFFFF'>";
    echo "<td>" . $row[""] . "</td> "; }
echo "</table>";

$db->close();
?>
```

เมื่อส่วนของ back-end เสร็จเรียบร้อยแล้ว จึงนำไฟล์ back-end ที่ได้ไปเรียกใช้งานในหน้า Front-end ในส่วนของ <body>.....</body> ด้วยโค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ 38 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
include('backend.php');
```



รูปที่ 3.16 หน้าเว็บทดสอบแสดงค่าเมื่อเชื่อมต่อฐานข้อมูลแล้ว

3.3.2 การทำ AJAX เพื่อให้แสดงข้อมูลแบบเรียลไทม์

โดยปกติหากไม่มีการเขียนโค้ด AJAX เข้าไป เมื่อข้อมูลมีการเปลี่ยนแปลง จำเป็นจะต้องกด F5 หรือรีเฟรชเว็บนั้นใหม่ทุกครั้ง แต่เมื่อต้องการแสดงค่าที่เพาเวอร์มิเตอร์วัดเข้ามาได้แบบเรียลไทม์ ไม่ต้องกดรีเฟรชเพื่ออัปเดตข้อมูล จึงจำเป็นต้องเขียนโค้ดในส่วนของ AJAX เข้าไปเพื่อตอบสนองความต้องการในส่วนนี้

การใช้ AJAX เพื่อดึงข้อมูลแบบเรียลไทม์หรือแสดงข้อมูลที่ Update ตลอดเวลาที่มีข้อมูลเข้ามาใหม่ในฐานข้อมูล ด้วย function ของ JavaScript

การใช้ JavaScript เข้ามาจัดการทำหน้าที่รีเฟรชข้อมูล ก็ถือเป็นวิธีที่สะดวกและยังใช้เทคโนโลยี AJAX เข้ามาช่วยให้โหลดข้อมูลเฉพาะบางส่วนนั้น ก็ช่วยให้การทำงานเร็วขึ้น และใช้ function ที่มีอยู่ของ JavaScript ให้ดึงข้อมูลทุก ๆ วินาที หรือ ระยะเวลาที่กำหนด หลักการที่ง่าย ๆ ก็คือใช้ function setTimeout() ในการทำงานเป็น Loop โดยเรียกข้อมูลซ้ำ ๆ ตามระยะเวลาที่กำหนด และภายใน Loop ก็จะมีการแทรก function ของ AJAX เพื่อไปโหลดข้อมูลมาแสดง และก็จะทำงานซ้ำ ๆ อย่างนี้ไปเรื่อย ๆ และในหน้าเว็บ เราก็จะได้ข้อมูลที่มีการ Update ล่าสุดเสมอ

โดยโค้ด AJAX จะทำงานอยู่ในส่วนของ Front-end เขียนโค้ดการทำงานได้ดังนี้

■ ในส่วนของ JavaScript

```
<script language="JavaScript">  
var HttpRequest = false;  
function doCallAjax(Sort) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

HttpRequest = false;
    if (window.XMLHttpRequest) {
        HttpRequest = new XMLHttpRequest();
        if (HttpRequest.overrideMimeType) {
            HttpRequest.overrideMimeType('text/html');
        }
    } else if (window.ActiveXObject) {
        try {
            HttpRequest = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
            try {
                HttpRequest = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (e) {
            }
        }
    }
    if (!HttpRequest) {
        alert('Cannot create XMLHttpRequest instance');
        return false;
    }

    var url = 'backend.php'; //ชื่อไฟล์ back-end ที่สร้างไว้
    var peters = 'mySort='+Sort;
    HttpRequest.open('GET',url,true);
    HttpRequest.setRequestHeader("Connection", "close");
    HttpRequest.onreadystatechange = function()
    {
        if(HttpRequest.readyState == 3) // Loading Request
        {
            document.getElementById("mySpan").innerHTML = "Now is Loading...";
        }
        if(HttpRequest.readyState == 4) // Return Request
        {
            document.getElementById("mySpan").innerHTML =
HttpRequest.responseText;
        }
    }
}

```

```
</script>
```

■ ในส่วนของ HTML

```
<body Onload="bodyOnload();">
<form name="frmMain" action="" method="post">
  <script language="JavaScript">
    function bodyOnload()
    {
      doCallAjax('Fre')
      setTimeout("doLoop();",1000); // 1000 = 1วินาที
    }
    function doLoop()
    {
      bodyOnload();
    }
  </script>
</body>
```

หลังจากเขียนโค้ดส่วน AJAX บนเว็บแล้ว เมื่อค่าในฐานข้อมูลถูกเปลี่ยน ค่าบนเว็บก็จะเปลี่ยนตามในทันทีโดยไม่ต้องกดรีเฟรชเว็บให้เสียเวลา เป็นการแสดงค่าแบบเรียลไทม์โดยมีเวลาในการที่เว็บจะดึงค่าใหม่ขึ้นมาโชว์ในทุก ๆ 1 วินาที

3.4 การเชื่อมต่อกับฐานข้อมูลของบอร์ด ASUS Tinker และนำมาแสดงผลแบบเรียลไทม์

เมื่อทดสอบการเชื่อมต่อฐานข้อมูลและนำมาแสดงผลแบบเรียลไทม์ได้แล้ว จึงนำโค้ดเรียลไทม์มาใช้ ซึ่งจากโค้ด AJAX ในที่เขียนไว้ นั้น จะเห็นว่ามีการใช้ HTTP request เรียกร้อยแล้ว เราจะนำโค้ด AJAX แบบเรียลไทม์ที่เขียนไว้มาแก้ไขเพื่อดึงข้อมูลจาก Web service ของบอร์ด ASUS Tinker ผ่าน Wi-Fi แทนการดึงข้อมูลจากฐานข้อมูลในเครื่องด้วย PHP แบบที่ทำในเว็บทดสอบ

เมื่อบอร์ด ASUS Tinker ได้ทำการเก็บข้อมูลจากเฟิร์มแวร์มีเตอร์มาลงในฐานข้อมูลบน SD Card เรียกร้อยแล้ว ตัวบอร์ดจะทำการส่งข้อมูลในฐานข้อมูลนั้นออกมาผ่าน Web service ในรูปแบบของ JSON ซึ่งมี Url เพื่อลิงค์เข้าสู่ Web service ได้ นั่นคือ <http://192.168.1.108:5000/datajson>

โดยเมื่อทางเว็บใช้ภาษา JavaScript ในการดึงข้อมูลขึ้นมาโชว์บนเว็บแบบเรียลไทม์จะทำการดึงเฉพาะค่าใน Array ที่ 0 ของทุกคอลัมน์ ในทุก ๆ 1 วินาที เพื่อให้ค่าที่โชว์บนเว็บเป็นค่าล่าสุดและอัปเดตข้อมูลอยู่ตลอดเวลา

```
[  
  {  
    "Current1": 0.0,  
    "Current2": 0.3,  
    "Current3": 0.0,  
    "Export_VArh": 1.65,  
    "Export_Wh": 0.11,  
    "Frequency": 50.02,  
    "Import_VArh": 0.3,  
    "Import_Wh": 7.3,  
    "L1_to_L2": 401.44,  
    "L2_to_L3": 403.81,  
    "L3_to_L1": 403.27,  
    "P1_LN_THD": 2.99,  
    "P2_LN_THD": 3.6,  
    "P3_LN_THD": 3.0,  
    "Power1": 0.0,  
    "Power2": 41.56,  
    "Power3": 0.0,  
    "Power_Factor1": 0.0,  
    "Power_Factor2": 0.92,  
    "Power_Factor3": 0.0,  
    "Tol_VAr": 0.0,  
    "Tol_VoltAmp": 41.56,  
    "Tol_power": 41.56,  
    "Tol_power_factor": -0.79,  
    "V1": 231.08,  
    "V2": 232.02,  
    "V3": 234.2,  
    "VAh_last_reset": 7.66,  
    "Volt_Amps1": 0.0,  
    "Volt_Amps2": 45.07,  
    "Volt_Amps3": 0.0,  
    "Volt_Amps_React1": 0.0,  
    "Volt_Amps_React2": 0.0,  
    "Volt_Amps_React3": 0.0,  
    "date_time": "2018-11-20 12:28:32"  
  },  
  {  
    "Current1": 0.0,  
    "Current2": 0.29,  
    "Current3": 0.0,  
    "Export_VArh": 1.65,  
  }  
]
```

รูปที่ 3.17 Web service ของบอร์ด ASUS Tinker

การดึงข้อมูลจาก Web service จะใช้ภาษา JavaScript ในการดึงข้อมูลออกมาแสดง โดยนำโค้ด AJAX จากหัวข้อ 3.3 มาแก้ไข

ในการเขียนโปรแกรมส่วนร้องขอข้อมูลจาก Web service จะต้องรับค่ามาเก็บไว้ในตัวแปรที่สร้างขึ้นใหม่ จากนั้นจึงค่อยเรียกตัวแปรเหล่านั้นมาแสดงบนหน้าเว็บ

ตารางที่ 3.1 ตัวแปรและพารามิเตอร์

กลุ่มพารามิเตอร์	ตัวแปร	รายละเอียด
	date_time	เวลาที่บันทึกค่า
พลังงานรวมของระบบไฟสามเฟส	Tol_power	กำลังไฟฟ้ารวมของระบบ (Watts)
	Tol_VoltAmp	กำลังไฟฟ้าปรากฏ (Apparent power) รวมของระบบ (VA)
	Tol_VAr	กำลังไฟฟ้ารีแอกทีฟ (Reactive power) รวมของระบบ (VAr)
	Tol_power_factor	Power factor รวมของระบบ
ค่าพารามิเตอร์พื้นฐาน	Current1	กระแสไฟฟ้าของเฟส 1 (Amps)
	Current2	กระแสไฟฟ้าของเฟส 2 (Amps)
	Current3	กระแสไฟฟ้าของเฟส 3 (Amps)
	V1	แรงดันไฟฟ้าของเฟส 1 (Volts)
	V2	แรงดันไฟฟ้าของเฟส 2 (Volts)
	V3	แรงดันไฟฟ้าของเฟส 3 (Volts)
	Power1	กำลังไฟฟ้าของเฟส 1 (Watts)
	Power2	กำลังไฟฟ้าของเฟส 2 (Watts)
	Power3	กำลังไฟฟ้าของเฟส 3 (Watts)
	Volt_Amps1	กำลังไฟฟ้าปรากฏของเฟส 1 (VA)
	Volt_Amps2	กำลังไฟฟ้าปรากฏของเฟส 2 (VA)
	Volt_Amps3	กำลังไฟฟ้าปรากฏของเฟส 3 (VA)
	Volt_Amps_React1	กำลังไฟฟ้ารีแอกทีฟของเฟส 1 (VAr)
	Volt_Amps_React2	กำลังไฟฟ้ารีแอกทีฟของเฟส 2 (VAr)
	Volt_Amps_React3	กำลังไฟฟ้ารีแอกทีฟของเฟส 3 (VAr)
	Power_Factor1	Power factor ของเฟส 1
	Power_Factor2	Power factor ของเฟส 2
	Power_Factor3	Power factor ของเฟส 3
	Frequency	ความถี่ของแหล่งจ่ายแรงดันไฟฟ้า (Hz)
พลังงานในแต่ละเฟส	L1_to_L2	Line1 to Line2 volts (V)
	L2_to_L3	Line2 to Line3 volts (V)
	L3_to_L1	Line3 to Line1 volts (V)

P1_LN_THD	Phase1 L/N volts of Total Harmonic Distortion (THD) (%)
P2_LN_THD	Phase2 L/N volts of Total Harmonic Distortion (THD) (%)
P3_LN_THD	Phase3 L/N volts of Total Harmonic Distortion (THD) (%)
Import_Wh	Import Wh since last reset (kWh/M or Wh)
Export_Wh	Export Wh since last reset (kWh/M or Wh)
Import_VArh	Import VArh since last reset (kVArh/MVArh)
Export_VArh	Export VArh since last reset (kVArh/MVArh)
VAh_last_reset	VAh since last reset (kVAh/MVAh)

- โค้ดเชื่อมต่อ Web service และดึงค่าจาก array ที่ 0 เพื่อแสดงค่าที่วัดได้ล่าสุด
 โดยโค้ดต่อไปนี้เป็นส่วนที่ได้แก้ไขในโค้ด AJAX ที่ได้เขียนเอาไว้ในหัวข้อ 3.3 หลักการการแก้ไขคือเปลี่ยนเส้นทางฐานข้อมูล จากฐานข้อมูลจำลองให้เป็น Url ของ Web service
 ใช้โค้ด `document.getElementById("...ชื่อ id ที่ตั้งขึ้นมาเอง...").innerHTML = obj[0]["...ชื่อในฐานข้อมูล.."];` ในการดึงข้อมูลจาก Web service มาเก็บไว้ใน id

```

<script language="JavaScript">
  if (!HttpRequest) {
    alert("Cannot create XMLHttpRequest");
    return false;
  }
  var url = "http://192.168.1.108:5000/datajson";
  var pmeters = 'mySort='+Sort;
  HttpRequest.open('GET',url,true);
  HttpRequest.onreadystatechange = function()
  {
    if(HttpRequest.readyState == 4) // Return Request
    {
      var a = this.responseText;
      var obj = JSON.parse(a);
      document.getElementById("Tol_power").innerHTML = obj[0]["Tol_power"];
      document.getElementById("date_time").innerHTML = obj[0]["date_time"];
      document.getElementById("Current1").innerHTML = obj[0]["Current1"];
      document.getElementById("V1").innerHTML = obj[0]["V1"];
    }
  }

```

```
}  
}  
</script>
```

- โค้ดแสดงผลบนหน้าเว็บ

เมื่อเชื่อมต่อกับ Web service ได้แล้ว จากนั้นจะนำค่าเหล่านั้นมาแสดงบนหน้าเว็บ ซึ่งจะแสดงออกมาในรูปแบบของตาราง โดยจะยกตัวอย่างการเขียนโค้ดแสดงผลแค่เมนู 'Total' ของเว็บเท่านั้น เพราะการแสดงผลในเมนูอื่น ๆ ยกเว้น History จะเขียนเหมือนกัน

โดยเขียนโค้ด HTML สร้างตารางขึ้นมา จากนั้นตกแต่งให้สวยงามด้วย CSS และในตารางจะชี้ไปที่ id ซึ่งได้ตั้งขึ้นมาเอง เพื่อเรียกแสดงค่าที่เก็บไว้ใน id นั้น

- โค้ด HTML

```
<center><table class="total1">  
  <thead>  
    <tr>  
      <th width='40%'>&nbsp;</th>  
      <th width='35%'>Value</th>  
      <th width='30%'>Unit</th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr> <td>Total system power</td>  
      <td id="Tol_power"></td>  
      <td>Watts</td></tr>  
    <tr> <td>Total system volt amps</td>  
      <td id="Tol_VoltAmp"></td>  
      <td>VA</td></tr>  
    <tr> <td>Total system VAR</td>  
      <td id="Tol_VAr"></td>  
      <td>VAr</td></tr>  
    <tr> <td>Total system power factor</td>  
      <td id="Tol_power_factor"></td>  
      <td>&nbsp;</td></tr>  
  </tbody>  
</tr>  
</table></center>
```

○ โค้ด CSS

```
table.total1 {
  background-color: #555555;
  text-align: center;
  border-collapse: collapse;
}
table.total1 td, table.total1 th {
  border: 1px solid #555555;
  padding: 5px 5px;
}
table.total1 tbody td {
  color: #FFFFFF;
}
table.total1 td:nth-child(even) {
  background: #398AA4;
}
table.total1 thead {
  background: #398AA4;
  border-bottom: 5px solid #398AA4;
}
table.total1 thead th {
  font-weight: bold;
  color: #FFFFFF;
  text-align: center;
  border-left: 2px solid #398AA4;
}
table.total1 thead th:first-child {
  border-left: none;
}
```

	Value	Unit
Total system power		Watts
Total system volt amps		VA
Total system VAr		VAr
Total system power factor		

รูปที่ 3.18 การแสดงผลเมนู Total

3.5 การแสดงค่าข้อมูลย้อนหลังโดยสามารถเลือกวันที่ได้

อีกหนึ่งความต้องการของลูกค้า นอกจากการแสดงผลแบบเรียลไทม์แล้ว ยังต้องสามารถเลือกวันที่เพื่อแสดงค่าพารามิเตอร์ย้อนหลังได้ ซึ่งบอร์ด ASUS Tinker ได้มี Web service ที่แสดงข้อมูลในรูปแบบของ JSON จะสามารถส่ง input วันที่เข้าไปเพื่อเชื่อมต่อเข้าสู่ Web service ของวันที่ตามที่กรอกลงไป โดยมี Url คือ

`http://192.168.1.108:5000/datajsonbydate?datetime=`

โดยที่หลังเท่ากับนั้นจะตามด้วยวันที่ที่ต้องการ มีรูปแบบคือ “ปี.ศ.-เดือน-วันที่” เช่น 2018-11-20

เราจะนำรูปแบบการกรอกวันที่ไปทำเป็นช่องค้นหาเพื่อให้ User สามารถกรอกวันที่ลงในช่องแล้วเชื่อมต่อเข้าสู่ข้อมูลในวันนั้น ๆ โดยใช้ภาษา JavaScript และ jQuery Data Tables มาช่วยในการทำตารางแสดงข้อมูลย้อนหลัง

```
[
  {
    "Current1": 0.0,
    "Current2": 0.32,
    "Current3": 0.0,
    "Export_VArh": 1.65,
    "Export_Wh": 0.11,
    "Frequency": 49.97,
    "Import_VArh": 0.3,
    "Import_Wh": 7.3,
    "L1_to_L2": 400.92,
    "L2_to_L3": 401.12,
    "L3_to_L1": 401.75,
    "P1_LN_THD": 2.63,
    "P2_LN_THD": 3.07,
    "P3_LN_THD": 2.75,
    "Power1": 0.0,
    "Power2": 39.18,
    "Power3": 0.0,
    "Power_Factor1": 0.0,
    "Power_Factor2": 0.96,
    "Power_Factor3": 0.0,
    "Tot_VAr": 0.0,
    "Tot_VoltAmp": 39.18,
    "Tot_power": 39.18,
    "Tot_power_factor": -0.79,
    "V1": 231.83,
    "V2": 231.1,
    "V3": 232.07,
    "VAh_last_reset": 7.66,
    "Volt_Amps1": 0.0,
    "Volt_Amps2": 40.9,
    "Volt_Amps3": 0.0,
    "Volt_Amps_React1": 0.0,
    "Volt_Amps_React2": 0.0,
    "Volt_Amps_React3": 0.0,
    "date_time": "2018-11-20 12:33:04"
  },
  {
    "Current1": 0.0,
    "Current2": 0.32,
    "Current3": 0.0,
    "Export_VArh": 1.65,
```

รูปที่ 3.19 ข้อมูลย้อนหลังบน Web service ของบอร์ด ASUS Tinker

3.5.1 การป้อน Input วันที่เพื่อเลือกหน้า web service

ขั้นแรกคือการสร้างเมนูบาร์หน้า History ขึ้นมาเพื่อใช้แสดงข้อมูลย้อนหลังบนเว็บ จากนั้นสร้างกล่องรับ Input เพื่อรับค่าวันที่เข้ามา และนำค่า Input นี้ไปต่อท้ายใน Path

<http://192.168.1.108:5000/datajsonbydate?datetime=> เพื่อเชื่อมเข้าหาข้อมูลในวันที่ที่ต้องการและนำมาแสดงด้วย Data Tables Plug-in ของ jQuery

ซึ่งโค้ดในการรับ Input และนำค่า Date Input ที่ได้รับเข้ามาไปต่อท้าย path ของ web service มีดังนี้

○ โค้ดกล่องรับ Input

```
YYYY-MM-DD : <input id="keyword" type="text"></input>
<button onclick="loadDoc()">Find</button>
```

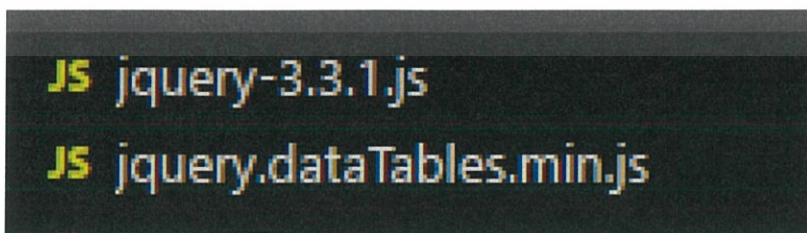
- โค้ดนำ Input ที่ได้รับไปต่อท้ายใส่ Path ของ Web service

```
<script type="text/javascript">
function loadDoc() {
    var key = document.getElementById('keyword');
    var baseUrl = "http://192.168.1.108:5000/datajsonbydate?datetime=";
    var date = key.value;
    var urls = baseUrl.concat(date);
    var objt = null;
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("mySpan").innerHTML = ' ';
        }
        else document.getElementById("mySpan").innerHTML = 'loading...';
    };
    xhttp.open("GET", urls, true);
    xhttp.send();
}
</script>
```

3.5.2 Data tables

การนำข้อมูลย้อนหลังในแต่ละวันมาแสดงนั้น ด้วยข้อมูลที่ทำให้การเก็บบันทึกไว้มีมากเกินไป จะใช้ JavaScript มาชี้ที่ละตัวเพื่อแสดงค่าได้ ดังนั้นจึงใช้ Data tables ซึ่งเป็น Plug-in ของ jQuery มาช่วยเพื่อความง่ายในการเขียนโค้ด

ซึ่งการใช้ jQuery นั้นจะต้องลง Library ของ jQuery ก่อนจึงจะใช้งานได้ โดยสามารถโหลด Library ได้จากเว็บไซต์ https://datatables.net/examples/data_sources/dom.html และเรียกใช้ในหน้าเว็บ



รูปที่ 3.20 Library ของ jQuery ที่โหลดมาจากเว็บ DataTables

โค้ดเรียกใช้ไลบรารี

```
<script type="text/javascript" src="jquery-3.3.1.js"></script>
<script type="text/javascript" src="jquery.dataTables.min.js"></script>
```

โค้ด DataTables สามารถนำมาจากเว็บไซต์ <https://datatables.net/> ได้เลย และนำมาแก้ไขเรียกไปที่ชื่อคอลัมน์ของฐานข้อมูลเราใน Web service ได้โค้ดดังต่อไปนี้

```
<script type="text/javascript" src="jquery-3.3.1.js"></script>
    if (this.readyState == 4 && this.status == 200) {
        document.getElementById("mySpan").innerHTML = ' ';
        var x = this.responseText;
        var objt = JSON.parse(x);
        $('#example').DataTable( {
            "data":objt,
            columns: [
                { data: 'date_time' },
                { data: 'Tot_power' },
                { data: 'Tot_VoltAmp' },
                { data: 'Tot_VAr' },
                { data: 'Tot_power_factor' },
            ]
        } );
    } else document.getElementById("mySpan").innerHTML = 'loading...';
};
xhttp.open("GET", urls, true);
xhttp.send();
}
</script>
```

นำข้อมูลที่เรียกใช้จาก Web service มาแสดงในตารางด้วยโค้ด HTML

```
<table id="example" class="display" style="width:100%">
    <thead>
        <tr>
            <th>Date Time</th>
            <th>Total power</th>
            <th>Total VA</th>
            <th>Total VAr</th>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<th>Total power factor</th>
</tr></thead>
</table>
```

โดยโค้ด CSS ของตารางสามารถโหลดมาจากเว็บ <https://datatables.net/> ได้เลย โดยสามารถคัดลอกนำมาวางในไฟล์ CSS ของเราที่ใช้สำหรับหน้าเว็บ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ51ษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

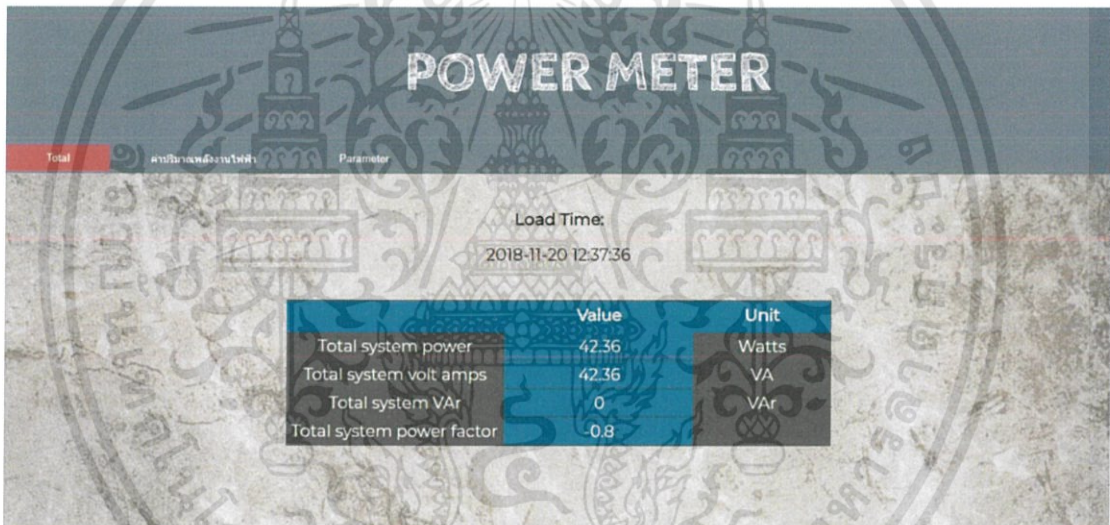
ผลการทดสอบโครงการ

4.1 กล่าวนำ

แสดงผลการดำเนินงานโครงการการแสดงผลข้อมูลการวัดพลังงานระยะไกลผ่านเว็บ โดยได้นำเพาเวอร์มิเตอร์ไปติดตั้งกับตู้ไฟของตึกสำนักงาน เพื่อวัดค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าจริง ๆ และนำมาแสดงผลบนเว็บ

4.2 การแสดงค่าแบบเรียลไทม์

เว็บสามารถแสดงค่าพารามิเตอร์แบบเรียลไทม์ได้ โดยหน้าเว็บจะทำการขอข้อมูลจาก Web service ทุก ๆ 1 วินาที เพื่อนำมาแสดงบนเว็บ โดยค่าบนหน้าเว็บจะถูกอัปเดตเองโดยไม่ต้องกดรีเฟรชเว็บ และเมื่อนำอุปกรณ์เพาเวอร์มิเตอร์ไปติดตั้งในตู้ไฟที่บริษัทเพื่อวัดค่าและเก็บข้อมูลจริง เว็บก็สามารถแสดงค่าได้ตรงตามกับค่าที่แสดงผลบนหน้าจอของเพาเวอร์มิเตอร์



รูปที่ 4.1 หน้าเว็บที่ 1 Total แสดงค่าพารามิเตอร์แบบเรียลไทม์

เว็บจะโชว์หน้าเว็บที่แสดงค่าพารามิเตอร์กลุ่มพลังงานรวมของระบบไฟสามเฟสเป็นหน้าแรก โดยในรูปภาพคือค่าพารามิเตอร์ที่วัดค่าจากตู้ไฟที่บริษัท

Total		ค่าปริมาณพลังงานไฟฟ้า		Parameter	
Load Time:					
2018-11-20 12:37:46					
	Voltage (Volts)	Current (Amps)	Power (Watts)		
L1	230.89	0	0		
L2	229.72	0.32	43.02		
L3	231.94	0	0		
	Volt amps (VA)	Volt amps reactive (VAr)	Power factor		
L1	0	0	0		
L2	44.81	0	0.96		
L3	0	0	0		
Frequency of supply voltages (Hz)					
49.95					

รูปที่ 4.2 หน้าเว็บที่ 2 ค่าปริมาณพลังงานไฟฟ้า แสดงค่าพารามิเตอร์แบบเรียลไทม์

หน้าเว็บที่สองของเว็บแสดงค่าพารามิเตอร์กลุ่มที่สอง คือกลุ่มพารามิเตอร์พื้นฐานเช่น แรงดันไฟฟ้า กระแสไฟฟ้า และ Power factor ของแต่ละเฟสเป็นต้น

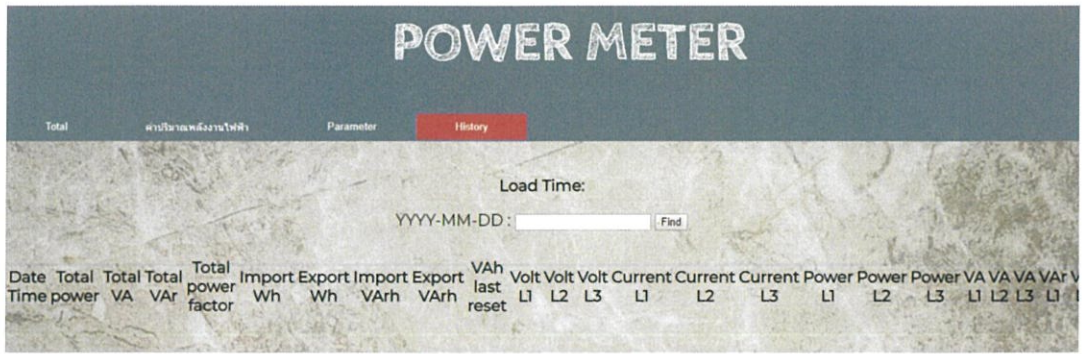
Load Time:			
2018-11-20 12:37:56			
	Value	Unit	
Line1 to Line2 volts	398.72	V	
Line2 to Line3 volts	399.7	V	
Line3 to Line1 volts	400.44	V	
Phase1 L/N volts THD	2.72	%	
Phase2 L/N volts THD	3.57	%	
Phase3 L/N volts THD	3.48	%	
	Value	Unit	
Import Wh since last reset	731	kWh/M or Wh	
Export Wh since last reset	0.11	kWh/M or Wh	
Import VArh since last reset	0.3	kVArh/MVArh	
Export VArh since last reset	1.65	kVArh/MVArh	
VArh since last reset	768	kVAh/MVAh	

รูปที่ 4.3 หน้าเว็บที่ 3 Parameter แสดงค่าพารามิเตอร์แบบเรียลไทม์

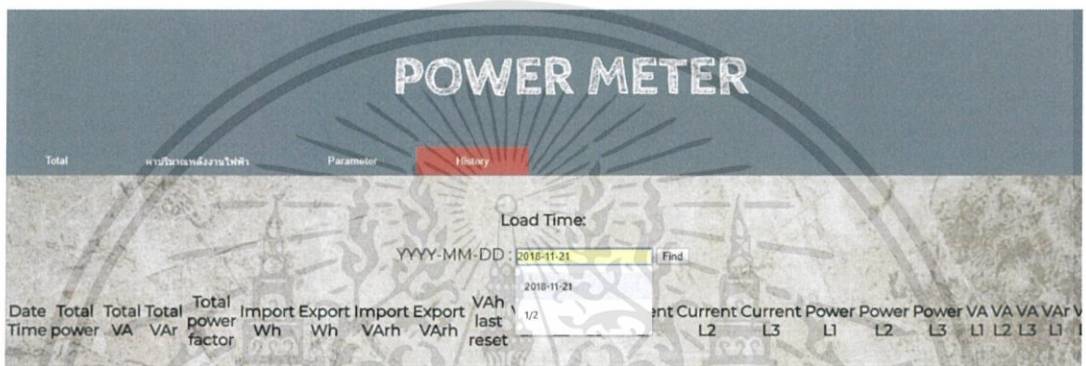
หน้าเว็บที่สามของเว็บคือการแสดงค่าพารามิเตอร์กลุ่มพลังงานของแต่ละเฟส ซึ่งค่าแต่ละค่าเป็นค่าจริงที่วัดได้จากการนำเอาเวอร์มิเตอร์ไปติดตั้งในตู้ไฟ และสามารถแสดงค่าแบบเรียลไทม์โดยอัปเดตค่าทุก ๆ 5 วินาที

4.3 การแสดงข้อมูลย้อนหลังโดยสามารถเลือกวันที่ที่ต้องการแสดงค่าย้อนหลังได้

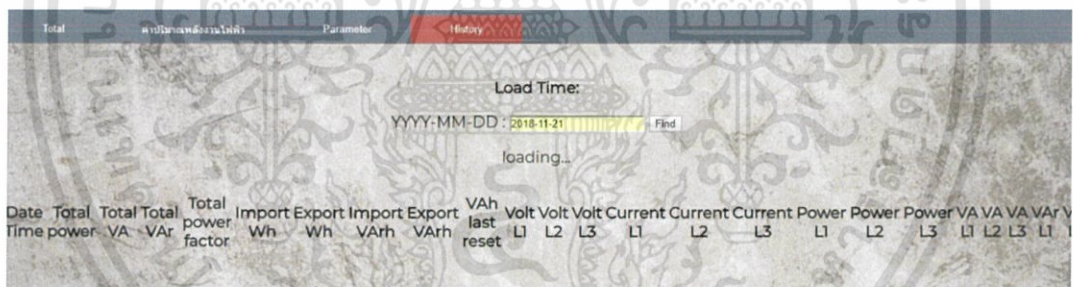
ผู้ใช้สามารถรอกวันที่ลงในช่องรับ Input ในรูปแบบ “ปี.ศ.-เดือน-วันที่” เช่น 2018-11-20 จากนั้นข้อมูลค่าพารามิเตอร์ในวันที่ที่กรอกลงไปนั้นจะถูกเรียกขึ้นมาแสดงในรูปแบบตาราง



รูปที่ 4.4 ช่องค้นหาวันที่ในเว็บ



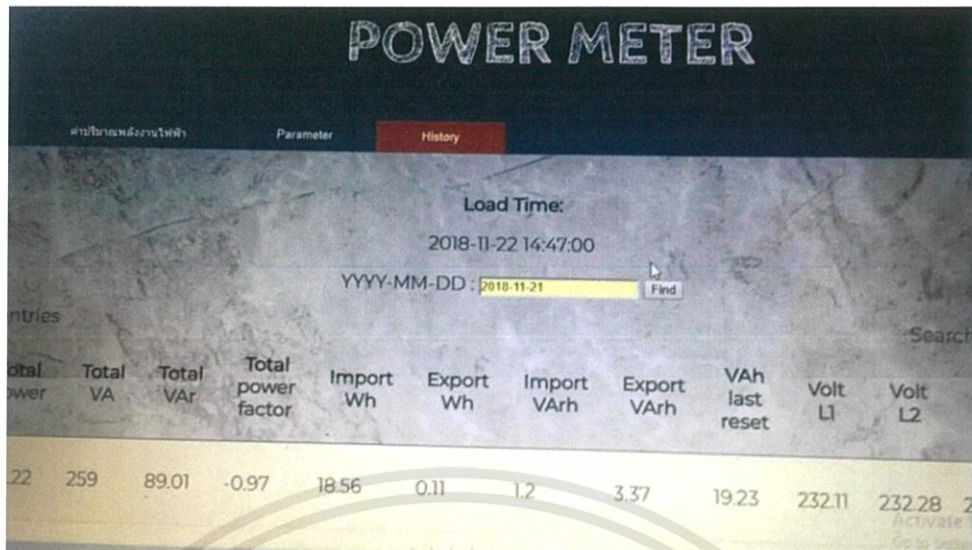
รูปที่ 4.5 การกรอกวันที่ลงในช่องค้นหา



รูปที่ 4.6 เมื่อกด Find จะขึ้น loading ระหว่างเรียกข้อมูล

เมื่อกดปุ่ม Find แล้ว บนหน้าเว็บจะแสดงคำว่า “loading...” ขึ้นระหว่างการดาวน์โหลดข้อมูลย้อนหลังเพื่อนำมาแสดงบนตาราง และเมื่อข้อมูลถูกโหลดเสร็จสิ้น ตัวอักษร “loading...” จะหายไป

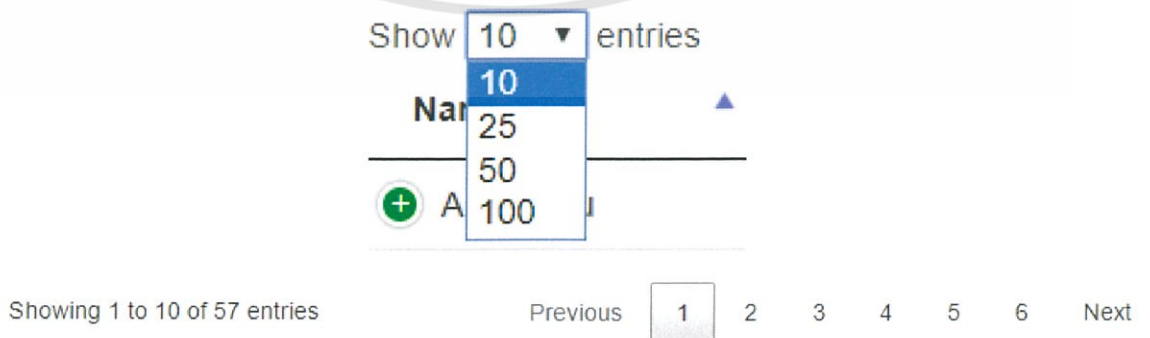
ข้อมูลย้อนหลังที่แสดงบนตารางนั้นจะแสดงข้อมูลค่าพารามิเตอร์และพลังงานไฟฟ้าทั้งหมดที่ได้ทำการบันทึกไว้ในแต่ละวัน ซึ่งมีค่าเป็นจำนวนมากทำให้อาจใช้เวลาในการดาวน์โหลดข้อมูลนาน



รูปที่ 4.7 เมื่อข้อมูลย้อนหลังโหลดเสร็จแล้ว

240.51	255.12	85.11	-0.98	18.56	0.11	1.2	3.37	19.23	231.5	233.44	236.32
240.77	255.03	85.77	-0.97	18.56	0.11	1.2	3.37	19.23	231.98	233.1	236.13
243.55	259.95	90.87	-0.97	18.56	0.11	1.2	3.37	19.23	232.75	232.7	235.77
243.68	260.05	90.8	-0.97	18.56	0.11	1.2	3.37	19.23	232.75	232.69	235.82
240.04	256.5	90.4	-0.97	18.56	0.11	1.2	3.37	19.23	232.69	232.78	235.73
242.82	259.2	90.67	-0.97	18.56	0.11	1.2	3.37	19.23	232.69	232.63	235.72
241.58	255.08	87.56	-0.97	18.56	0.11	1.22	3.37	19.23	232.38	233.8	235.46

รูปที่ 4.8 ข้อมูลย้อนหลังแบบเลือกวันที่ในตาราง



รูปที่ 4.9 ฟังก์ชันตารางข้อมูลย้อนหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงข้อมูลจะตัดย่อยข้อมูลออกเป็นหลาย ๆ หน้าตาราง สามารถคลิกเลือกได้ว่าจะแสดงข้อมูลในหน้าตารางละกี่ Entries



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ 56 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการดำเนินการร่วมกับบริษัทเสก้า ออโตเมชั่น จำกัด เพื่อพัฒนาเพาเวอร์มิเตอร์ให้สามารถอ่านค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าได้จากระยะไกล เป็นการอำนวยความสะดวกและลดภาระค่าใช้จ่ายจากการเดินทางไปที่โรงงานเพื่อเก็บบันทึกข้อมูลจากเพาเวอร์มิเตอร์ โดยนำบอร์ด ASUS Tinker มาช่วยเป็นสื่อการในการเก็บบันทึกข้อมูลและส่งออกสู่เว็บหรือโปรแกรมแสดงผลที่สร้างขึ้น

โรงงานสหกิจศึกษาฉบับนี้ได้ทำการสร้างในส่วนของเว็บสำหรับแสดงผลข้อมูลการวัดพลังงานจากระยะไกล โดยได้เขียนเว็บขึ้นมาจากภาษา HTML, CSS และ JavaScript ให้สามารถแสดงผลข้อมูลค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าทั้งสามกลุ่มออกมาในรูปแบบของตาราง โดยการแสดงผลนั้นจะเป็นแบบเรียลไทม์และแบบเรียกดูข้อมูลย้อนหลังด้วยการกรอกวันที่ จากนั้นนำเพาเวอร์มิเตอร์ไปติดตั้งที่ตู้ไปในสถานประกอบการเพื่อวัดค่าพารามิเตอร์และปริมาณพลังงานไฟฟ้าของไฟสามเฟสเพื่อตรวจสอบค่าที่แสดงผลแบบเรียลไทม์และตรวจสอบการเรียกอ่านข้อมูลย้อนหลังบนเว็บว่าสามารถทำได้จริง

5.2 ปัญหาและวิธีการแก้ไข

5.2.1 ปัญหาที่พบ

1. ความรู้พื้นฐานเกี่ยวกับภาษาคอมพิวเตอร์และการเขียนเว็บไม่มากพอ เนื่องจากเป็นองค์ความรู้ที่ไม่ตรงตามหลักสูตรที่เรียนมา ทำให้ต้องเริ่มศึกษาค้นคว้าด้วยตนเอง และจากความรู้ไม่ชำนาญทำให้โค้ดที่เขียนมีความซับซ้อนโดยไม่จำเป็น

2. ความรู้พื้นฐานเกี่ยวกับภาษาต่าง ๆ และ Plug-in ที่สามารถอำนวยความสะดวกในการเขียนโปรแกรมไม่มากพอ ทำให้เกิดปัญหาเขียนโค้ดที่ยากและยาว และเมื่อต้องแก้ไขทำให้เป็นไปได้ยาก จึงต้องลบโปรแกรมส่วนนั้นและเขียนใหม่โดยศึกษาภาษาอื่นเพิ่มขึ้นมาอีก

3. ความต้องการฟังก์ชันเว็บเปลี่ยนแปลงอยู่บ่อยครั้ง ต้องการแสดงค่าพารามิเตอร์ที่มากขึ้นและเพิ่มเติมให้มีการแสดงค่าแบบย้อนหลัง ทำให้เกิดการรีอโปรแกรมใหม่ทั้งหมดเพราะโปรแกรมเก่าไม่สามารถแก้ไขให้หน้าเว็บเป็นไปตามความต้องการได้

5.2.2 วิธีการแก้ไข

1. จากปัญหาเรื่ององค์ความรู้พื้นฐานต่าง ๆ ในการเขียนโปรแกรม และปัญหาเมื่อตัวโปรแกรมเกิดมีข้อผิดพลาดขึ้น จึงได้มีการขอคำแนะนำจากโปรแกรมเมอร์ในบริษัทเพื่อรับคำชี้แนะและวิธีการแก้ปัญหที่เกิดขึ้นกับตัวโปรแกรม

2. จากปัญหาเรื่องความต้องการฟังก์ชันของเว็บที่เปลี่ยนแปลงบ่อย เมื่อทำงานเสร็จสิ้นในแต่ละส่วนจึงนำผลงานเข้าไปให้พี่นิเทศน์งานตรวจสอบ เพื่อจะได้แก้ไขหรือเพิ่มเติมได้ทันก่อนเริ่มทำงานในส่วนต่อไป

5.3 ข้อเสนอแนะ

ในการออกแบบเว็บไซต์สำหรับแสดงผล ควรลำดับขั้นตอนการเขียนโค้ดและแบ่งออกเป็น ส่วน ๆ อย่างชัดเจน เพื่อให้ง่ายต่อการแก้ไขและพัฒนาในอนาคต รวมถึงควรศึกษาภาษาโปรแกรมในปัจจุบันให้มากขึ้น เพราะมีภาษาจำนวนมากถูกพัฒนาเพิ่มขึ้นมาเพื่อให้ง่ายต่อการใช้งาน ทำให้สามารถลดปัญหาความซับซ้อนและยุ่งยากของตัวโปรแกรมได้



เอกสารอ้างอิง

- [1] XAMPP คืออะไร แหล่งที่มา: <https://www.ninetechno.com/a/website/873-xampp.html>
- [2] Apache คืออะไร แหล่งที่มา: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2265-apache-คืออะไร.html>
- [3] Tinker board แหล่งที่มา: <https://www.asus.com/th/Motherboards/Tinker-Board/>
- [4] รู้จักกับ Visual Studio Code แหล่งที่มา: <https://www.mindphp.com/บทความ/microsoft/4829-visual-studio-code.html>
- [5] ระบบจัดการฐานข้อมูล แหล่งที่มา: <https://th.wikipedia.org/wiki/ระบบจัดการฐานข้อมูล>
- [6] HTML และ CSS แหล่งที่มา: <https://cmuis51.wordpress.com/2011/05/02/html-css/>
- [7] ภาษา HTML เบื้องต้น แหล่งที่มา: <http://internship09.wmu.psru.ac.th/2017/01/13/basic-html/>
- [8] CSS เบื้องต้น แหล่งที่มา: <http://www.kontentblue.com/site/article/article.php?id=css-introduction&page=1>
- [9] โครงสร้างของภาษา PHP แหล่งที่มา: <http://marcuscode.com/lang/php/program-struct>
- [10] AJAX เบื้องต้น แหล่งที่มา: <https://goragod.com/index.php?module=knowledge&id=399>
- [11] Ajax Introduction แหล่งที่มา: <https://www.thaicreate.com/tutorial/ajax-introduction.html>
- [12] พื้นฐาน JavaScript เบื้องต้น แหล่งที่มา: <https://medium.com/@worawit422/พื้นฐาน-javascriptเบื้องต้น-8976595b907a>
- [13] jQuery แหล่งที่มา: <https://www.thaicreate.com/jquery/jquery-what-is-it.html>
- [14] Data Tables แหล่งที่มา: <https://datatables.net/>
- [15] Tutorial Code แหล่งที่มา: <https://www.w3schools.com>
- [16] Real-Time Ajax แหล่งที่มา: <https://www.thaicreate.com/community/jquery-get-load-data-real-time.html>
- [17] ระบบไฟสามเฟส แหล่งที่มา: http://www.tngroup.co.th/media/article_detail/157