

## บทที่ 2

### เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการจัดทำระบบจองคิวสอบโปรเจกออนไลน์ โดยพัฒนาระบบระบบจองคิวสอบโปรเจกออนไลน์ เพื่อให้บรรลุวัตถุประสงค์ของการทำปริญญานิพนธ์ในครั้งนี้ ผู้จัดทำจะต้องศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องดังนี้

- 2.1 ทฤษฎีการทำระบบจองคิว
- 2.2 ระบบฐานข้อมูล
- 2.3 ทฤษฎีการพัฒนาโปรแกรม
- 2.4 โปรแกรมที่ใช้ในการพัฒนาระบบ
- 2.5 งานวิจัยที่เกี่ยวข้อง
- 2.6 การประเมินความพึงพอใจ

#### 2.1 ทฤษฎีการทำระบบจองคิว

การจองคิว คือ การลงทะเบียนนัดหมายล่วงหน้าเพื่อจุดประสงค์ใดประสงค์หนึ่งโดยที่ผู้นัดและผู้รับนัด ต้องมีเวลาว่างตรงกันและไม่สามารถนัดกับบุคคลที่สามในวันเวลาดังกล่าวที่ได้ทำการนัดไปแล้ว โดยการจองคิวจะมีทั้งแบบ “ออฟไลน์” และ “ออนไลน์”

##### 2.1.1 การจองคิวแบบ “ออฟไลน์”

การจองคิวแบบ “ออฟไลน์” คือ การที่ผู้ใช้บริการต้องเข้าไปทำการลงทะเบียนเพื่อนัดหมายล่วงหน้ายังสถานที่ที่ให้บริการต่าง ๆ แบบตัวต่อตัว มีการบันทึกข้อมูลเป็นลายลักษณ์อักษรลงบนกระดาษ เก็บไว้ทั้งสองฝ่าย เพื่อเป็นหลักฐานในการยืนยันว่ามีการลงทะเบียนนัดพบจริง ยกตัวอย่างเช่น นายเอได้ไปที่โรงแรมบีบีคิว เพื่อทำการจองโต๊ะอาหารสำหรับมื้อค่ำ ซึ่งทางโรงแรมบีบีคิวจะขอให้นายเอทำการลงทะเบียนเพื่อยืนยันการจอง เมื่อลงทะเบียนเสร็จนายเอจะได้ข้อมูลโดยย่อ เช่นชื่อโรงแรม เวลาที่ทำการจอง และหมายเลขโต๊ะมาเป็นกระดาษ เพื่อให้ให้นายเอนำมาแจ้งพนักงานเมื่อมาถึงในมื้อค่ำ ดังที่กล่าวมาข้างต้น จะเป็นการลงทะเบียนจองคิวนัดหมายแบบ “ออฟไลน์”

### 2.1.2 การจองคิวแบบ “ออนไลน์”

การจองคิวแบบ “ออนไลน์” คือ การนัดหมายล่วงหน้าโดยใช้การทำงานบนอินเทอร์เน็ตสามารถทำงานจากที่ใดก็ได้ เมื่อมีอินเทอร์เน็ต ไม่จำเป็นต้องเข้าไปยังสถานที่นั้น ๆ เพื่อทำการนัดหมายหรือจองคิว ซึ่งในปัจจุบันไม่ว่าจะเป็นภาครัฐหรือเอกชนเริ่มมีการนำระบบการจองคิวแบบออนไลน์เข้ามาใช้งาน เช่น ระบบบัตรคิวออนไลน์\_SEE-Q ของ โรงพยาบาลชัยนาท, จองคิวพาสปอร์ตออนไลน์, จองคิวทำใบขับขี่ออนไลน์ เป็นต้น โดยการที่นำระบบจองคิวออนไลน์เข้ามาใช้งาน ทำให้ลดการทำงานและระยะเวลาจากเดิมที่ต้องเดินเรื่องเอกสารขอติดต่อเป็นเวลานานให้ลดลง สะดวกทั้งผู้ใช้บริการและพนักงาน

## 2.2 ระบบฐานข้อมูล

### 2.2.1 องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูลโดยทั่วไปจะเกี่ยวกับ 4 ส่วนหลัก ๆ ดังนี้

2.2.1.1 ข้อมูล (Data) คือ ข้อมูลที่จัดเก็บอยู่ในฐานข้อมูล ข้อมูลสามารถที่จะใช้งานร่วมกันได้ และผู้ใช้สามารถเรียกใช้ข้อมูลพร้อมกันได้

2.2.1.2 ฮาร์ดแวร์ (Hardware) คือ ส่วนของอุปกรณ์ที่เกี่ยวข้องกับระบบฐานข้อมูล ได้แก่ หน่วยความจำหลัก (Memory) เป็นตัวที่นำข้อมูลจากฐานข้อมูลขึ้นมาประมวลผล และหน่วยเก็บบันทึกข้อมูลภายนอก หรือหน่วยความจำสำรอง (Secondary Storage) เป็นตัวที่จัดเก็บข้อมูลของฐานข้อมูล

2.2.1.3 ซอฟต์แวร์ (Software) คือ เป็นโปรแกรมที่ช่วยจัดการควบคุมความถูกต้อง ความเข้าซ้องของข้อมูล และความสัมพันธ์ต่าง ๆ ภายในฐานข้อมูล ส่งผลให้ผู้ใช้สามารถที่จะเรียกข้อมูลได้โดยไม่จำเป็นต้องต้องทราบถึงโครงสร้างทางกายภาพของข้อมูลและอำนวยความสะดวกให้แก่ผู้ใช้ในการใช้ฐานข้อมูล

2.2.1.4 ผู้ใช้ระบบฐานข้อมูล (User) คือ ผู้ที่เรียกใช้ข้อมูลจากระบบฐานข้อมูล สามารถแบ่งออกเป็น 3 กลุ่ม ดังนี้

ก) ผู้พัฒนาโปรแกรม (Application Programmer) คือ ผู้ที่ทำหน้าที่พัฒนาโปรแกรม เพื่อเรียกใช้ข้อมูลจากระบบฐานข้อมูลมาประมวลผล

ข) กลุ่มผู้ใช้ (End User) คือ ผู้ที่นำข้อมูลจากฐานข้อมูลไปใช้งาน ซึ่งแบ่งออกเป็น 2 กลุ่ม ดังนี้

- 1) Naive User ได้แก่ ผู้ใช้ที่เรียกใช้ข้อมูลโดยอาศัยโปรแกรมที่พัฒนาขึ้น
- 2) Sophisticated User ได้แก่ ผู้ใช้ที่เรียกใช้ข้อมูลจากฐานข้อมูลด้วยประโยคคำสั่ง คือ คำสั่งที่ใช้ดึงข้อมูลใน Database

ค) Database Administrator (DBA) ได้แก่ ผู้บริหารที่ทำหน้าที่ควบคุมและตัดสินใจในการกำหนดโครงสร้างของฐานข้อมูล ชนิดข้อมูล วิธีการจัดเก็บข้อมูล รูปแบบในการเรียกใช้ข้อมูล ความปลอดภัยของข้อมูลและกฎระเบียบที่ใช้ควบคุมความถูกต้องของข้อมูลภายในฐานข้อมูล โดยอาศัยคำสั่งในกลุ่ม Data Definition Language (DDL) ซึ่งเป็นส่วนหนึ่งของ Query Language เป็นตัวกำหนด

## 2.2.2 โครงสร้างข้อมูล

ในการจัดเตรียมข้อมูลเข้าสู่ขั้นตอนการประมวลผลนั้น ข้อมูลจะต้องได้รับการจัดใหม่อยู่ในรูปแบบที่เครื่องคอมพิวเตอร์สามารถรับได้ คือ การจัดการโครงสร้างของข้อมูลซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้

2.2.2.1 ฟิลด์ (Filed) คือ กลุ่มข้อมูลตัวหนังสือ ตัวเลข หรือสัญลักษณ์พิเศษต่าง ๆ ที่มีความสัมพันธ์กัน และแสดงลักษณะหรือความหมายอย่างใดอย่างหนึ่ง โดยทั่วไปฟิลด์สามารถแบ่งได้ 3 ประเภท ดังนี้

ก) ฟิลด์ตัวเลข (Number Field) คือ ฟิลด์ที่เป็นกลุ่มของตัวเลข จำนวนเต็ม จำนวนเต็มบวก จำนวนเต็มลบ

ข) ฟิลด์ตัวอักษร (Alphabetic Field) คือ เป็นกลุ่มของตัวอักษรที่เป็นตัวอักษรหรือช่องว่างระหว่างตัวอักษร

ค) ฟิลด์อักขระ (Alphanumeric) คือ เป็นกลุ่มของตัวอักขระที่เป็นตัวเลขหรือตัวอักษร

2.2.2.2 เรคคอร์ด (Record) คือ ชนิดข้อมูลที่สามารถเก็บข้อมูลชนิดอื่น ๆ ไว้ภายในได้โดยเราเรียกข้อมูลแต่ละตัวที่อยู่ภายในว่า ฟิลด์ (Filed) และก่อนที่จะใช้งานเรคคอร์ดได้นั้นจะต้องประกาศชนิดของเรคคอร์ดเสียก่อน จากนั้นทำการประกาศค่าตัวแปร เรคคอร์ด หรือระเบียบ คือ กลุ่มของฟิลด์ที่มีความสัมพันธ์กันในรูปแบบใดรูปแบบหนึ่งเรคคอร์ด ประกอบด้วยฟิลด์ต่างประเภทกันเป็นชุด โดยพื้นฐานของเรคคอร์ดจะต้องมีฟิลด์ที่ใช้อ้างอิงอย่างน้อย 1 ฟิลด์ซึ่งเรียกว่า คีย์ฟิลด์ (Key Field) และฟิลด์ที่จะใช้เป็นคีย์ฟิลด์ในแต่ละเรคคอร์ดจะต้องไม่ซ้ำกัน

2.2.2.3 ไฟล์ (File) หรือแฟ้มข้อมูล คือ กลุ่มของเรคคอร์ดที่มีความสัมพันธ์กันในด้านใดด้านหนึ่ง ดังนั้นไฟล์จึงประกอบด้วย เรคคอร์ดหลาย ๆ เรคคอร์ดมารวมกัน

2.2.2.4 ฐานข้อมูล (Database) คือ ฐานข้อมูลประกอบด้วยไฟล์หรือแฟ้มข้อมูลที่มีความสัมพันธ์กันโดยใช้หลักการไม่ให้มีข้อมูลซ้ำกัน สามารถเรียกใช้งานได้อย่างถูกต้องและรวดเร็ว ซึ่งเรียกว่าระบบการจัดการฐานข้อมูล

### 2.2.3 ระบบการจัดการฐานข้อมูล (Database Management System : DBMS)

เป็นโปรแกรมที่ทำหน้าเป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อนและความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูล ซึ่งต่างจากระบบฐานข้อมูล หน้าที่เหล่านี้จะเป็นหน้าที่ของโปรแกรมเมอร์ ในการติดต่อกับฐานข้อมูลในฐานข้อมูล ไม่ว่าจะด้วยการใช้คำสั่ง DML หรือ DDL หรือจะด้วยโปรแกรมต่าง ๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูลจะถูกโปรแกรม DBMS นำมาแปล (Compile) เป็นการกระทำ (Operation) ต่าง ๆ ภายใต้อำนาจนั้น เพื่อนำไปกระทำตัวข้อมูลในฐานข้อมูลต่อไป สำหรับส่วนการทำงานต่าง ๆ ภายในโปรแกรม DBMS ที่ทำหน้าที่ในการแปลคำสั่งไปเป็นการกระทำต่าง ๆ ที่จะกระทำกับข้อมูลประกอบด้วยส่วนการทำงานต่าง ๆ ดังนี้

2.2.3.1 Database Manager ส่วนที่ทำหน้าที่กำหนดการกระทำต่าง ๆ ให้กับส่วน File Manager เพื่อไปกระทำกับข้อมูลที่เก็บอยู่ในฐานข้อมูล (File Manager เป็นส่วนที่ทำหน้าที่บริหารและจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลในระดับกายภาพ)

2.2.3.2 Query Processor เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของ Query Language ให้อยู่ในรูปแบบของคำสั่งที่ Database Manager เข้าใจ

2.2.3.3 Data Manipulation Language Precompiled (DML) เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DML ให้อยู่ในรูปแบบที่ส่วน Application Programs Object Code จะนำไปเข้ารหัสเพื่อส่งต่อไปยังส่วน Database Manager ในการแปลประโยคคำสั่งของกลุ่มคำสั่ง DML ของส่วน Data Manipulation Language Precompiled นี้จะต้องทำงานร่วมกับส่วน Query Processor

2.2.3.4 Data Definition Language Precompiled (DDL) เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DDL ให้อยู่ในรูปแบบของ Data Dictionary ของฐานข้อมูล (Metadata) ได้แก่ รายละเอียดที่บอกถึงโครงสร้างต่าง ๆ ของข้อมูล

2.2.3.5 Application Programs Object Code เป็นส่วนที่ทำหน้าที่แปลงคำสั่งต่าง ๆ เป็นโปรแกรม รวมทั้งคำสั่งในกลุ่มคำสั่ง DML ที่ส่งต่อมาจาก Data Definition Language Precompiled ให้อยู่ในรูปของ Object Code ที่จะส่งต่อไปให้ Database Manager เพื่อกระทำกับข้อมูลในฐานข้อมูล (โอภาส เอี่ยมสิริวงศ์, 2545)

#### 2.2.3.6 หน้าที่หลักของระบบจัดการฐานข้อมูล (DBMS-Functional)

ก) รักษาความถูกต้องของข้อมูล (Data Integrity) ป้องกันไม่ให้มีการนำข้อมูลที่ขัดแย้งกัน (Data Redundancy) ข้อมูลที่ไม่ถูกต้องเข้าสู่ฐานข้อมูลซึ่งในการควบคุมความถูกต้องของข้อมูลจะขึ้นอยู่กับกฎ (Constraint) ที่ผู้ออกแบบฐานข้อมูลที่กำหนดขึ้นมา

ข) จัดการดูแลควบคุมการเข้าถึงข้อมูลในฐานข้อมูล (Data Access) เช่น การเรียกดูข้อมูล (Select) การบันทึกข้อมูล (Insert) การแก้ไขข้อมูล (Update) ซึ่งคำสั่งหรือภาษาที่ใช้ติดต่อกับระบบจัดการฐานข้อมูลเพื่อเข้าถึงข้อมูลได้แก่ Structured Query Language (SQL)

ค) จัดการดูแลระบบรักษาความปลอดภัย (Security) และจัดการระดับสิทธิ (Permission) การเข้าถึงข้อมูลในฐานข้อมูลของผู้ใช้แต่ละราย เช่น การเรียกดูข้อมูลได้ (Select) เพิ่มข้อมูลได้ (Insert) แก้ไขข้อมูลได้ (Update) เป็นต้น

ง) จัดการดูแลการใช้งานข้อมูลในฐานข้อมูลร่วมกันจากผู้ใช้ หรือโปรแกรมมากกว่าหนึ่ง (Multi User Accessibility) เพื่อให้ผู้ใช้หรือโปรแกรมได้ข้อมูลที่ต้องการอยู่เสมอ

จ) จัดการดูแลบำรุงรักษาข้อมูลในฐานข้อมูล (Data Maintenance) กรณีข้อมูลเกิดความเสียหายสามารถซ่อมแซม และเรียกข้อมูลกลับคืนมาได้ (Data Repair and Recovery)

### 2.2.4 ประโยชน์ของฐานข้อมูล

2.2.4.1 สามารถลดความซ้ำซ้อนของข้อมูล (Data Redundancy) โดยไม่จำเป็นต้องจัดเก็บข้อมูลที่ซ้ำซ้อนกันไว้ในระบบแฟ้มข้อมูลของแต่ละหน่วยงานเหมือนเช่นเดิม แต่สามารถนำข้อมูลมาใช้ร่วมกันในคุณลักษณะ Integrated แทน

2.2.4.2 สามารถหลีกเลี่ยงความขัดแย้งของข้อมูล (Data Inconsistency) เนื่องจากไม่ต้องจัดเก็บข้อมูลที่ซ้ำซ้อนกันในหลายแฟ้มข้อมูล

2.2.4.3 แต่ละหน่วยงานในองค์กรสามารถใช้ข้อมูลร่วมกันได้

2.2.4.4 สามารถกำหนดให้ข้อมูลมีรูปแบบที่เป็นมาตรฐานเดียวกันได้ เพื่อให้ผู้ใช้ข้อมูลในฐานข้อมูลชุดเดียวกัน สามารถเข้าใจและสื่อสารถึงความหมายเดียวกัน

2.2.4.5 สามารถกำหนดระบบความปลอดภัยให้กับฐานข้อมูลได้ โดยกำหนดระดับความสามารถในการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนให้แตกต่างกันตามความรับผิดชอบ

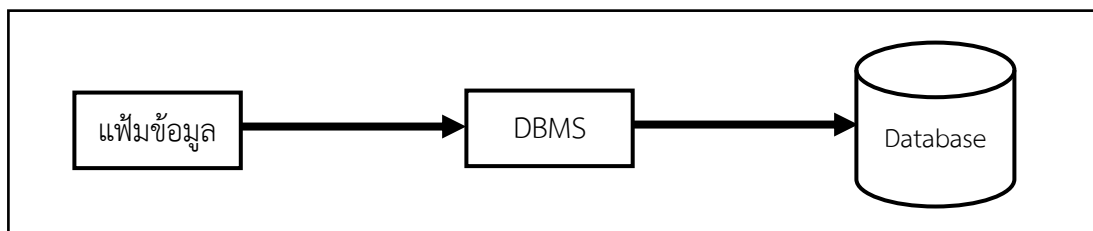
2.2.4.6 สามารถรักษาความถูกต้องของข้อมูลได้โดยการระบุกฎเกณฑ์ในการควบคุมความผิดพลาดที่อาจเกิดขึ้นจากการป้อนข้อมูลผิดได้

2.2.4.7 สามารถตอบสนองต่อความต้องการใช้ข้อมูลในหลายรูปแบบ

2.2.4.8 ทำให้ข้อมูลเป็นอิสระจากโปรแกรมที่ใช้งานข้อมูลนั้น (Data Independence)

### 2.1.5 ระบบฐานข้อมูลกับคอมพิวเตอร์

ฐานข้อมูล คือ แหล่งที่ใช้เก็บรวบรวมข้อมูล ซึ่งข้อมูลที่ถูกเก็บรวบรวมไว้นี้จะถูกจัดการเพื่อตอบสนองความต้องการของผู้สร้างฐานข้อมูลในการสร้างฐานข้อมูลมีความเป็นที่จะต้องแจกแจงข้อมูลที่ใช้ในระบบงานพร้อมทั้งตั้งชื่อข้อมูลแต่ละตัว ชื่อที่ตั้งควรเป็นมาตรฐานและมีเพียงชื่อเดียวเพราะผู้ใช้แต่ละคนจำเป็นต้องอ้างอิงถึงข้อมูล โดยการตั้งชื่อที่เหมาะสมจะสามารถหลีกเลี่ยงการสับสนการเรียกใช้ข้อมูลได้ การจัดเก็บฐานข้อมูลเป็นการนำเอาฐานข้อมูลมาจัดเก็บรวมกันภายใต้ข้อมูลเดียวกัน ซึ่งสามารถใช้ข้อมูลและสามารถแก้ไขปัญหาที่เกิดขึ้นในระบบเพิ่มข้อมูลได้ ข้อมูลต่าง ๆ ที่ถูกจัดเก็บในฐานข้อมูลจะเป็นฐานข้อมูลที่สัมพันธ์กันและเป็นข้อมูลที่สนับสนุนการดำเนินงานขององค์กร จึงกล่าวได้ว่าฐานข้อมูลแต่ละตัวจะเทียบเท่ากับเพิ่มข้อมูล 1 ระบบ และจะเรียกข้อมูลที่สร้างขึ้นเพื่อสนับสนุนการดำเนินงานอย่างใดอย่างหนึ่งนั้นว่า “ระบบฐานข้อมูล” (Database System)



ภาพที่ 2-1 แสดงสัญลักษณ์ฐานข้อมูล

#### 2.2.5.1 อธิบายคำศัพท์จาก E-R Diagrams (Entity-Relationship Diagrams)

ก) เอนทิตี (Entity) เป็นรูปภาพที่ใช้แทนสิ่งที่เป็นรูปธรรมของสิ่งต่าง ๆ ที่สามารถระบุได้ในความเป็นจริง ซึ่งอาจเป็นสิ่งที่จับต้องได้ เช่น บุคคล สิ่งของ

ข) แอททริบิวต์ (Attributes) เป็นสิ่งที่ใช้อธิบายคุณลักษณะของเอนทิตีหนึ่ง ๆ ซึ่งมีความหมายเดียวกันกับฟิลด์หรือเขตข้อมูล

ค) ความสัมพันธ์ (Relationships) ใช้แสดงความสัมพันธ์ระหว่างแต่ละเอนทิตีในเอนทิตีจะแสดงโดยการใช้สัญลักษณ์สี่เหลี่ยมข้าวหลามตัดแทนความสัมพันธ์ สำหรับสัญลักษณ์ที่ใช้แทนแอททริบิวต์จะใช้รูปวงรีโดยมีเส้นเชื่อมไปยังเอนทิตี

2.2.5.2 ประเภทของความสัมพันธ์ระหว่างเอนทิตี ความสัมพันธ์ระหว่างเอนทิตีเป็นความสัมพันธ์ที่สมาชิกของเอนทิตีหนึ่งสัมพันธ์กับสมาชิกของเอนทิตี ซึ่งสามารถแบ่งประเภทของความสัมพันธ์ออกเป็น 2 ประเภท คือ

##### ก) ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-One)

จะใช้สัญลักษณ์ 1:1 แทนความสัมพันธ์แบบหนึ่งต่อหนึ่งซึ่งความสัมพันธ์แบบนี้จะเป็นความสัมพันธ์ที่สมาชิกหนึ่งรายการของเอนทิตีมีความสัมพันธ์กับสมาชิกหนึ่งรายการของอีกเอนทิตีหนึ่งเช่น สามี 1 คน มีภรรยาโดยตามกฎหมายได้ 1 คน และภรรยา 1 คน มีสามีโดยตามกฎหมายได้ 1 คน

##### ข) ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-Many)

จะใช้สัญลักษณ์ 1 : M แทนความสัมพันธ์แบบหนึ่งต่อกลุ่ม ซึ่งความสัมพันธ์แบบนี้จะเป็นความสัมพันธ์ที่สมาชิกหนึ่งรายการของเอนทิตีมีความสัมพันธ์กับสมาชิกหลายรายการในเอนทิตีหนึ่ง เช่น นิสิต 1 คน สามารถมีอาจารย์ที่ปรึกษาได้เพียง 1 ท่าน อาจารย์แต่ละท่านสามารถเป็นอาจารย์ที่ปรึกษานิสิตได้หลายคน

##### ค) ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-Many)

จะใช้สัญลักษณ์ M : M แทนความสัมพันธ์แบบกลุ่มต่อกลุ่ม ซึ่งความสัมพันธ์แบบนี้จะเป็นความสัมพันธ์ที่สมาชิกหนึ่งรายการของเอนทิตี มีความสัมพันธ์กับสมาชิกหลายรายการในเอนทิตีหลายรายการ เช่น สินค้าหลายรายการสามารถอยู่ในใบสั่งซื้อได้หลายใบ ใบสั่งซื้อมีสินค้าหลายรายการ

2.2.5.3 รูปแบบของข้อมูล การจัดการฐานข้อมูลในปัจจุบันสามารถจัดประเภทของรูปแบบข้อมูล Data Model ได้ 3 รูปแบบ ดังนี้

ก) ฐานข้อมูลแบบลำดับชั้น (Hierarchical Model) จะมีความสัมพันธ์ระหว่างแฟ้มข้อมูลเป็นลำดับชั้นอาวโส แฟ้มข้อมูลจะมีตำแหน่งจากบนลงล่าง โดยที่แฟ้มข้อมูลที่อยู่ในระดับสูงกว่าจะเป็นแม่ของแฟ้มข้อมูลที่อยู่ต่ำกว่า ซึ่งจะมีข้อสังเกตว่าลำดับชั้นหนึ่ง ๆ แฟ้มข้อมูลหนึ่งจะมีแฟ้มข้อมูลย่อยได้เพียงแฟ้มเดียว

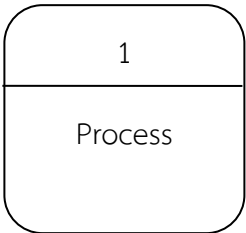
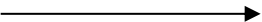
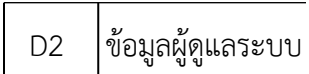

ข) ฐานข้อมูลแบบเครือข่าย (Network Model) จะมีลักษณะคล้ายกับแบบฐานข้อมูลแบบลำดับชั้น แต่มีข้อมูลแตกต่างตรงที่ฐานข้อมูลแบบเครือข่ายสามารถมีแฟ้มข้อมูลหลักได้มากกว่าหนึ่งแฟ้ม

ค) ฐานข้อมูลแบบความสัมพันธ์ (Relational Model) มีลักษณะที่แตกต่างจากฐานข้อมูลทั้ง 2 แบบแรก กล่าวคือ จะไม่มีแฟ้มข้อมูลหลักหรือแฟ้มข้อมูลลูก คือ แฟ้มข้อมูลแต่ละส่วนจะเป็นอิสระต่อกันและไม่มีความสัมพันธ์กันโดยใช้องค์ประกอบข้อมูล การที่ฐานข้อมูลแบบนี้จะถูกเรียกใช้เป็นฐานข้อมูลสัมพันธ์ด้วยเหตุผลที่ว่าข้อมูลจะสร้างความสัมพันธ์ของตัวเองขึ้นมา

2.2.5.4 ทฤษฎีเกี่ยวกับแผนภาพกระแสข้อมูล (Data Flow Diagram : DFD) เป็นภาพแสดงการเปลี่ยนแปลงของข้อมูลในขณะไหลผ่านกระบวนการทำงานต่าง ๆ ของระบบสารสนเทศ DFD จึงเป็นภาพหรือเป็นโครงสร้างของระบบงานสารสนเทศที่สื่อให้เข้าใจการทำงานของระบบงานในรูปแบบของความสัมพันธ์ระหว่างกระแสข้อมูลและกระบวนการ DFD ไม่ได้สื่อความหมายในลักษณะที่ตรรกะ (Logic) ของกระบวนการทำงาน กล่าวโดยง่าย ๆ ภาพ DFD ทำให้เราเข้าใจส่วนประกอบของระบบงาน เข้าใจการใช้ข้อมูลในแต่ละกระบวนการและข้อมูลที่เป็นผลจากการทำงานของกระบวนการโดยโครงสร้างจะเริ่มจากระดับสูงสุดซึ่งจะแสดงส่วนที่อยู่ภายนอกกระบวนการ ส่วนนี้สำคัญเพราะเป็นส่วนที่บอกว่าจะระบบนั้น ๆ ได้รับข้อมูลมาจากที่ใดและผลลัพธ์ต่าง ๆ ถูกส่งไปที่ใดบ้าง DFD ในระดับลึกลงไปจะไม่แสดงสิ่งที่อยู่นอกกระบวนการ คือ ไม่มีสิ่งนั้นเป็นส่วนประกอบ โดยปกติหรือถ้าเป็นไปได้ จะวางแหล่งที่มาของข้อมูลไว้ทางซ้ายมือของ DFD และส่วนภายนอกที่รับผลลัพธ์ของระบบจะอยู่ทางขวามือ ทั้งนี้เพื่อให้อยู่ในรูปแบบของกระแสข้อมูลจากซ้ายไปขวา แต่หลาย ๆ กรณีไม่อาจเรียงตามแนวทางนี้ได้ เนื่องจากบางครั้งอินพุตและผลลัพธ์อาจจะเป็นสิ่งเดียวกัน ในกรณีนั้นเราจะวางข้อมูล และผลลัพธ์ไว้ในที่ที่เหมาะสมซึ่งอาจจะอยู่เหนือกระบวนการหรือใต้กระบวนการก็ได้ DFD ระดับรองลงมา (Low-Level Data Flow Diagram) คือ ส่วนที่แสดงระบบย่อยลงมาจาก DFD ที่กล่าวมาแล้วหรือเรียกว่า ระดับแม่ เมื่อระดับแม่ไม่สามารถแสดงรายละเอียดทั้งหมดได้เป็นต้องแตก Level ย่อยออกมาเพื่อแสดงการประมวลขั้นตอนการทำงานให้ชัดเจนขึ้น โดยสัญลักษณ์ที่ใช้ในแผนภาพกระแสข้อมูลแสดงได้ดังตารางที่ 2-1



ตารางที่ 2-1 สัญลักษณ์ที่ใช้ในภาพกระแสข้อมูล (DFD)

สัญลักษณ์	ชื่อสัญลักษณ์และคำอธิบาย
	<p>โพรเซส (Process) มีหน้าที่รับข้อมูล และทำการ คำนวณ เรียบเรียง เปลี่ยนสภาพของข้อมูลทำให้เกิดข้อมูลชุดใหม่ โดยจะเขียนชื่อ Process ไว้ในวงกลม การตั้งชื่อ Process ให้สื่อหลักดังนี้ นำหน้าด้วยคำ กิริยา และตามด้วยคำนามที่สื่อความหมายของ Process นั้น ๆ</p>
	<p>กระแสข้อมูล (Data Flow) แสดงส่วนของข้อมูลที่ถูกส่งเข้า กระบวนการประมวลผลและผลลัพธ์ที่ได้ ผ่านขบวนการประมวลแล้ว ทุก Process ที่อยู่ใน DFD จะต้องมีการรับและส่งข้อมูลเข้าและออก จาก Process เสมอ</p>
	<p>ที่เก็บข้อมูล (Data Store) คือ แหล่งเก็บข้อมูลซึ่งอยู่ภายนอกของ Process</p>
	<p>สิ่งภายนอก (External Entity) คือ สิ่งที่อยู่นอกระบบประมวลผลข้อมูล อาจหมายถึงบุคคล หน่วยงาน ระบบประมวลผลอื่นที่มีหน้าที่ส่งข้อมูลให้หรือรับข้อมูลจาก DFD ของระบบงาน</p>

## 2.3 ทฤษฎีการพัฒนาโปรแกรม

### 2.3.1 ทฤษฎีการวิเคราะห์และออกแบบระบบ

การวิเคราะห์ระบบ (System Analysis) เป็นการศึกษา วิเคราะห์และแยกแยะถึงปัญหาที่เกิดขึ้นในระบบ พร้อมทั้งเสนอแนะทางเสนอแนะทางแก้ไขตามความต้องการของผู้ใช้งานและความเหมาะสมต่อสถานะทางการเงินขององค์กร การออกแบบระบบ (System Design) คือ การสร้างแบบพิมพ์เขียวของระบบใหม่ตามความต้องการในเอกสารความต้องการระบบ กำหนดสิ่งที่จำเป็น เช่น อินพุต เอาท์พุท ส่วนต่อประสานผู้ใช้และการประมวลผล เพื่อประกันความน่าเชื่อถือ ความถูกต้องแม่นยำ การบำรุงรักษาได้และความปลอดภัยของระบบ นอกจากนั้นการออกแบบระบบเป็นวิธีการออกแบบและกำหนดคุณสมบัติทางเทคนิคโดยนำระบบคอมพิวเตอร์มาประยุกต์ใช้ เพื่อแก้ปัญหาที่ทำการวิเคราะห์มาแล้ว ขั้นตอนการวิเคราะห์และออกแบบระบบออกเป็น 2 ระดับคือ ขั้นต้น และขั้นสูง

#### 2.3.1.1 ขั้นต้น (Basic System Analysis) ประกอบด้วย 8 ขั้นตอนดังนี้

ก) System Requirement เป็นการรับทราบปัญหา หรือความต้องการของผู้ใช้ หรือเจ้าของงานอาจเรียกรวมได้ว่า เป็นขั้นตอนของการเก็บรายละเอียด

ข) Context Description เป็นการกำหนดบริบท ประกอบด้วย List of Entities, List of Data และ List of Process

ค) Context Diagram เป็นการออกแบบโครงสร้างบริบท โดยอาศัยข้อมูลในขั้นตอนที่ ข) นักวิเคราะห์ระบบบางราย มีความถนัดที่จะทำขั้นตอนนี้ก่อนขั้นตอนที่ ข) ซึ่งไม่มีผลเสียแต่อย่างใด

ง) Process Hierarchy Chart เป็นการเขียนผังการไหลของข้อมูลในระดับต่าง ๆ ที่ปรากฏตามขั้นตอนที่ ค)

จ) Data Flow Diagram : DFD เป็นการเขียนผังการไหลของข้อมูลในระดับต่าง ๆ ที่ปรากฏตามขั้นตอนที่ ง)

ฉ) Process Description เป็นการอธิบายรายละเอียดกระบวนการให้ชัดเจนขึ้น โดยทั่วไปนิยมอธิบายใน End Process ของแต่ละ Root

ช) Data Modeling เป็นขั้นตอนการกำหนด Cardinality เพื่อพิจารณาความสัมพันธ์ของ Entities ทั้งหมดที่เกิดขึ้นในระบบ ซึ่งใช้ Data Storage ที่ได้ในขั้นตอน DFD

ซ) Data Dictionary เป็นขั้นตอนกำหนด Attribute ที่อ้างถึงใน Data Modeling เพื่อกำหนดรายละเอียดที่จะเป็นเบื้องต้นสำหรับใช้ในระบบ

### 2.3.1.2 ขั้นสูง (Advance System Analysis) ประกอบ 4 ขั้นตอนคือ

ก) Database Design เป็นขั้นตอนการออกแบบฐานข้อมูล โดยอาศัยข้อมูลนำเข้าในขั้นที่ 2.3.1.1.(ข) และ 2.3.1.1.(ค) ซึ่งอาจใช้วิธีการ Normalization หรือ Entity Relationship Model แล้วแต่กรณี ซึ่งไม่จำเป็นว่าจะต้องได้ Normal Form (5NF) ขึ้นอยู่กับ นักวิเคราะห์ระบบจะเห็นว่า มีความจำเป็นและเหมาะสมในระดับใด แต่ทั้งนี้ควรไม่ต่ำกว่า Boyce Codd Normal Form (BCNF)

ข) Data Table Description เป็นขั้นตอนกำหนดรายละเอียด Attribute ที่มีในแต่ละ Table โดยอาศัยข้อมูลจากขั้นตอนที่ 2.3.1.1.(ข) และ 2.3.1.2.(ก)

ค) Output Design หรือ การออกแบบส่วนแสดงผล แยกออกเป็นรายงาน เอกสารและ ข้อความมีพฤติกรรม 3 ชนิด

- 1) แสดงผลจากฐานข้อมูลโดยตรง (Data to Output)
- 2) แสดงผลจากการประมวลผลที่ได้รับจากการข้อมูลนำเข้า (Data-Process to Output)
- 3) แสดงผลโดยตรงจากข้อมูลนำเข้า (Input to Output)

โดยสามารถแสดงผลได้ทั้งกระดาษและจอภาพ การออกแบบ Output Design ควรกระทำก่อนการ ออกแบบอื่น ๆ ทั้งหมด เพราะจะช่วยตรวจสอบว่า มี Attribute ที่ออกแบบไว้ในขั้น 2.2 ครบถ้วนหรือไม่

ง) Input Design หรือการออกแบบส่วนนำข้อมูลเข้า วัตถุประสงค์เป็นการออกแบบเพื่อนำข้อมูลเข้าไปในระบบคอมพิวเตอร์ จึงถูกออกแบบให้มีรูปแบบสอดคล้องกับการแสดงผลทางจอภาพ คือ 25 บรรทัด 80 คอลัมน์ แม้ว่าบางครั้งจะถูกออกแบบเป็นแบบบันทึกข้อมูลล่วงหน้า ก่อนนำมาบันทึกผ่านจอภาพ ก็ยังอ้างอิงกับตำแหน่งทางจอภาพ เพื่อหลีกเลี่ยงความสับสนของผู้ใช้ แบ่งออกเป็น 2 พฤติกรรม ดังนี้

- 1) ออกแบบฟอร์มเอกสารกรอกข้อมูล
- 2) ออกแบบส่วนติดต่อกับผู้ใช้ ซึ่งมี 3 ชนิด คือ ส่วนติดต่อกับผู้ใช้ด้วยเมนู ด้วยคำสั่ง และด้วยกราฟิก

### 2.3.2 วงจรการพัฒนาระบบ

วงจรการพัฒนาระบบ (System Development Life Cycle : SDLC) คือ กระบวนการทางความคิด (Logical Process) ในการพัฒนาระบบสารสนเทศเพื่อแก้ปัญหาทางธุรกิจและตอบสนอง ความต้องการของผู้ใช้ได้ โดยระบบที่จะพัฒนานั้น อาจเริ่มด้วยการพัฒนาระบบใหม่เลยหรือนำระบบเดิมที่มีอยู่ แล้วมาปรับเปลี่ยนให้ดียิ่งขึ้น ภายในวงจรนี้จะแบ่งกระบวนการพัฒนาออกเป็นระยะ (Phases) ได้แก่

ระยะการวางแผน (Planning Phase) ระยะการวิเคราะห์ (Analysis Phase) ระยะการออกแบบ (Design Phase) และระยะการสร้างและพัฒนา (Implementation Phase) โดยแต่ละระยะจะประกอบไปด้วย ขั้นตอน (Steps) ต่าง ๆ แตกต่างกันไปตาม Methodology ที่นักวิเคราะห์นำมาใช้ เพื่อให้เหมาะสมกับสถานะทางการเงินและความพร้อมขององค์กรในขณะนั้น ขั้นตอนในวงจรพัฒนาระบบช่วยให้นักวิเคราะห์ระบบสามารถดำเนินการได้อย่างมีแนวทางและเป็นขั้นตอน ทำให้สามารถควบคุมระยะเวลาและงบประมาณในการปฏิบัติงานของโครงการพัฒนาระบบได้ ขั้นตอนต่าง ๆ นั้นมีลักษณะคล้ายกับการตัดสินใจแก้ปัญหาตามแนวทางวิทยาศาสตร์ (Scientific Management) อัน ได้แก่ การค้นหาปัญหา การค้นหาแนวทางแก้ไขปัญหา การประเมินผลแนวทางแก้ไขปัญหาที่ค้นพบ เลือกแนวทางที่ดีที่สุด และพัฒนาทางเลือกนั้นให้ใช้งานได้ สำหรับวงจรการพัฒนาระบบในหนังสือเล่มนี้ จะแบ่งเป็น 7 ขั้นตอน ดังนี้

#### 2.3.2.1 การวิเคราะห์ปัญหา (Problem Analysis)

- ก) เป็นการวิเคราะห์ปัญหาของระบบงานเดิม
- ข) เมื่อองค์กรมีความต้องการที่จะสร้างระบบสารสนเทศขึ้น
- ค) เป็นหน้าที่ของผู้จัดการโครงการ (Project Manager) ผู้ออกแบบฐานข้อมูล (Database Analysis) นักวิเคราะห์ระบบ (System Analysis) และต้องร่วมกันทำงาน

#### 2.3.2.2 ศึกษาความเป็นไปได้ (Feasibility Study)

- ก) ความเป็นไปได้ของเทคโนโลยี (Technology Feasibility)
  - 1) ระบบงานเดิมมีอุปกรณ์ทางด้านฮาร์ดแวร์ และซอฟต์แวร์เพียงพอที่จะรองรับสารสนเทศที่จะเกิดขึ้นหรือไม่
  - 2) ถ้าไม่เพียงพอต้องวิเคราะห์ว่าจะจัดซื้อฮาร์ดแวร์ และซอฟต์แวร์ประเภทใดเพิ่มเติม
  - 3) ถ้ามีอยู่แล้วต้องวิเคราะห์ว่ามีความสามารถเพียงพอหรือไม่
- ข) ความเป็นไปได้ทางด้านการปฏิบัติการ (Operational Feasibility)
  - 1) บุคลากรเดิมมีความสามารถหรือประสบการณ์ในการพัฒนาและติดตั้งระบบหรือไม่
  - 2) ผู้ใช้ระบบมีความคิดเห็นอย่างไรกับการเปลี่ยนแปลงของระบบที่จะเกิดขึ้น

ค) ความเป็นไปได้ทางด้านเศรษฐศาสตร์ (Feasibility Study)

- 1) ศึกษาถึงค่าใช้จ่ายต่าง ๆ ที่จะเกิดขึ้นตั้งแต่เริ่มต้นด้านพัฒนาระบบไปจนกระทั่งมีการติดตั้ง และใช้งานระบบจริง รวมถึงค่าใช้จ่ายประจำวัน
- 2) คาดการณ์ถึงผลประโยชน์ที่จะได้รับ
- 3) เวลาที่ใช้ในการพัฒนาระบบ
- 4) ตัดสินใจว่าเปลี่ยนแปลงระบบหรือไม่

2.3.2.3 วิเคราะห์ความต้องการของผู้ใช้ (User Requirement Analysis)

- ก) ศึกษาระบบการทำงานเดิมให้เข้าใจ
- ข) กำหนดขอบเขตของฐานข้อมูลที่จะสร้างขึ้น
- ค) กำหนดความสามารถของโปรแกรมประยุกต์ที่จะสร้างขึ้น
- ง) กำหนดอุปกรณ์ทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่จะมีใช้ การวางแผนระยะเวลาในการทำงาน

2.3.2.4 การออกแบบฐานข้อมูล (Database Design)

- ก) การวิเคราะห์หาเอนทิตีหรือรีเลชัน
- ข) การวิเคราะห์ความสัมพันธ์

2.3.2.5 การออกแบบและพัฒนาโปรแกรม (Implementation)

- ก) จะมีการเลือกระบบจัดการฐานข้อมูลขึ้นมาใช้
- ข) ออกแบบโปรแกรมว่าระบบจะต้องประกอบด้วยโปรแกรมอะไรบ้าง แต่ละโปรแกรมมีหน้าที่อะไร มีความสัมพันธ์กันอย่างไรการเชื่อมโยงระหว่างโปรแกรมทำอย่างไร
- ค) ออกแบบหน้าจอการนำข้อมูลเข้า รูปแบบรายงาน การควบคุม คงสภาพข้อมูล
- ง) เพื่อนำข้อมูลที่ได้มาสร้างเป็นเอกสารการออกแบบโปรแกรม (Program - Specification) ส่งให้โปรแกรมเมอร์ในการเขียนโปรแกรมต่อไป
- จ) ในขั้นตอนการพัฒนาโปรแกรม โปรแกรมเมอร์ต้องทำการเขียนและทดสอบความถูกต้องของโปรแกรม (Program Testing)
- ฉ) หลังจากนั้นก็นำโปรแกรมมารวมกันและทำการทดสอบอีกทีเรียกว่าการทดสอบระบบ (System Testing)
- ช) ทำการทดสอบและแก้ไขจนกระทั่งระบบมีความถูกต้องตามต้องการ

### 2.3.2.6 ทำเอกสารประกอบโปรแกรม (Documentation)

ก) เป็นการอธิบายรายละเอียดของโปรแกรมว่าจุดประสงค์ของโปรแกรมคืออะไร ใช้งานในด้านไหน ฯลฯ

ข) สรุปรายละเอียดของโปรแกรมและแสดงเป็นผังงาน (Flowchart) หรือรหัสจำลอง (Pseudo code) ก็ได้

ค) เอกสารประกอบ มีอยู่ 2 แบบ ดังนี้

1) โปรแกรมสำหรับผู้ใช้ (User Documentation) สำหรับผู้ที่ไม่เกี่ยวข้องกับการพัฒนาโปรแกรม แต่เป็นผู้ที่ใช้งานโปรแกรมอย่างเดียว

2) เอกสารประกอบโปรแกรมสำหรับผู้เขียนโปรแกรม (Technical-Documentation)

### 2.3.2.7 การติดตั้งและบำรุงรักษาโปรแกรม (Program Maintenance)

ก) เมื่อโปรแกรมผ่านการตรวจสอบตามขั้นตอนเรียบร้อยแล้ว จะถูกนำมาติดตั้งให้ใช้งาน

ข) รวมถึงการฝึกอบรมให้กับพนักงานที่ต้องใช้จริง เพื่อให้ทำงานได้โดยไม่มีปัญหาซึ่งในระยะแรก ๆ อาจมีปัญหาในการใช้งานได้

ค) ระยะแรกจะต้องมีผู้คอยดูแลและตรวจสอบการทำงานและเมื่อมีการใช้งานไปนาน ๆ อาจต้องมีการแก้ไขปรับปรุงโปรแกรมให้เหมาะกับเหตุการณ์และความต้องการของผู้ใช้ที่เปลี่ยนแปลงไป

## 2.4 โปรแกรมที่ใช้ในการพัฒนาระบบ

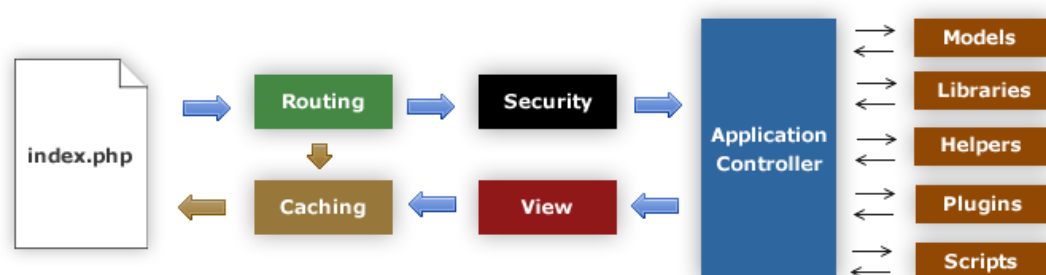
### 2.4.1 CodeIgniter

CodeIgniter คือ framework ที่ถูกพัฒนาขึ้นด้วย ภาษา PHP สำหรับ CodeIgniter Framework เป็นเครื่องมือสำหรับช่วยพัฒนาเว็บไซต์และเว็บแอปพลิเคชัน โดยมีโครงสร้างการพัฒนาโปรแกรมอย่างเป็นระบบและรวมคำสั่งต่างๆ ที่จำเป็นต่อการพัฒนาเว็บไซต์ด้วย PHP ไว้ภายใน CodeIgniter พัฒนาโดย Rick Ellis ซึ่งมีลิขสิทธิ์เป็น Open Source สามารถดาวน์โหลดและใช้ได้ฟรี ถูกพัฒนาขึ้นตั้งแต่ปี 2006 และยังมีการพัฒนามาเรื่อยๆ จนถึงปัจจุบัน (CodeIgniter คืออะไร, 2562)

2.4.1.1 โครงสร้างของ CodeIgniter Structure CodeIgniter รองรับการเขียนแบบ Model-View-Controller (MVC) เป็นการแยกส่วนประมวลผล ออกจากส่วนแสดงผล ทำให้โครงสร้างมีความปลอดภัยมากยิ่งขึ้น และง่ายต่อการพัฒนาโปรแกรม โค้ดที่ได้มีความเป็นระเบียบและง่ายต่อการนำไปแก้ไข CodeIgniter ยังออกแบบให้สามารถติดต่อกับฐานข้อมูลชนิดต่างๆ ได้ง่ายอีกด้วย เช่น mysql, postgres, sqlserver หรือ Oracle เป็นต้น หากเรียกใช้ฐานข้อมูลถูกต้องตามหลักของ database library แล้ว ผู้ใช้สามารถเปลี่ยนชนิดของฐานข้อมูลได้ เพียงแค่แก้ค่า config ส่วนระบบ url ของระบบ codeigniter สามารถนำไปพัฒนาได้ทั้ง เว็บไซต์ และเว็บแอปพลิเคชัน ไฟล์ภายใน CodeIgniter ประกอบด้วย โครงสร้างหลักๆ คือ application, system, index.php ระบบภายในสามารถแยกโปรเจคออกเป็น หลายๆ application ได้

#### ตารางที่ 2-2 โครงสร้างไฟล์ภายในโฟลเดอร์ application

cache	เป็นส่วนเก็บข้อมูลที่ประมวลผลแล้ว เพื่อการเรียกใช้ครั้งต่อไป
config	ใช้ตั้งค่าระบบต่างๆ ภายใน application
controllers	เป็นส่วนรวมโค้ดการประมวลผล จะเขียนโค้ด PHP เป็นส่วนใหญ่
core	เป็นส่วนเสริมที่มีไว้สำหรับการใช้งาน (MVC) ขึ้นอีกชั้นหนึ่ง ในชื่อ modules
helpers	หากต้องการเขียนฟังก์ชันใช้เองในระบบสามารถสร้างและเก็บไว้ในที่นี้
hooks	ใช้สำหรับสร้างไฟล์ เพื่อเข้าไป เพิ่ม, แก้ไข Class ต่างๆ ตามเราต้องการ
language	เป็นที่เก็บไฟล์ภาษาต่างๆ สำหรับผู้ที่ต้องการทำเว็บหลายภาษา
libraries	เป็นที่เก็บไฟล์ Class ที่คุณเขียนขึ้นเอง หากต้องการเรียกใช้ Class อื่นๆ ต้องเรียก get_instance ก่อน
logs	เป็นที่เก็บไฟล์บันทึกประวัติการทำงานภายใน application
models	คล้ายๆ libraries แต่ models จะสามารถเรียกใช้ libraries ที่คุณโหลดเข้ามาในระบบได้เลย
third_party	สำหรับเก็บโฟลเดอร์ ที่จะนำไปเรียกใช้งานใน libraries
views	ส่วนนี้เป็นการแสดงผล ส่วนใหญ่เราจะเขียนโค้ด html ที่นี่ แต่นามสกุลไฟล์ต้องเป็น .php เท่านั้น



ภาพที่ 2-2 Model-View-Controller MVC

ตารางที่ 2-3 อธิบายโฟลเดอร์เก็บไฟล์ของ Model-View-Controller MVC

Model-View-Controller MVC	คือ โครงสร้างตามหลักการ ระบบจะแยกส่วนประมวลผล ออกจากส่วนแสดงผล แต่ในทางปฏิบัติ สามารถเขียนโค้ด PHP ลงในส่วนแสดงผลได้เช่นกัน
Models	คือ การอธิบายโครงสร้างข้อมูล แบบฉบับของคลาสแบบจำลอง จะมีฟังก์ชันที่ช่วยให้รับ , ใส , ปรับปรุงข้อมูลในฐานข้อมูล
View	คือ ข้อมูลที่ถูกแสดงให้กับผู้ใช้ View ส่วนใหญ่จะเป็นเว็บเพจ แต่ใน CodeIgniter , view สามารถเป็นเพียงบางส่วนของหน้าได้ เช่น ส่วนหัว ส่วนท้าย มันสามารถจะเป็นหน้า RSS หรือเป็นอย่างอื่นได้ที่เป็น "เพจ"
Controller	คือ ส่วนที่ถูกใช้เป็นสื่อกลาง ระหว่าง Model , View และแหล่งที่มาอื่นๆ ที่ต้องการในการทำงาน HTTP request และการสร้างเว็บเพจ



### 2.4.1.2 ขั้นตอนการทำงาน CodeIgniter

โดย CodeIgniter จะแบ่งการทำงานออกเป็นส่วนย่อย ๆ โดยระบบจะแยกส่วนประมวลผล ออกจากส่วนแสดงผล เพื่อให้ง่ายต่อการใช้งานและสะดวกรวดเร็ว ไม่หนัก

ตารางที่ 2-4 ขั้นตอนการทำงานของ Model-View-Controller MVC

index.php	เป็นตัวควบคุมส่วนหน้า, สร้างทรัพยากรพื้นฐานที่ต้องการในการรัน CodeIgniter
Routing	ตัว Router ทำการตรวจสอบ HTTP request กำหนดว่าควรจะทำอะไรกับมัน
Caching	ถ้ามีไฟล์แคชอยู่ ตัวมันจะถูกส่งกลับทันทีไปยังเบราว์เซอร์ โดยไม่ผ่านการทำงานปกติของระบบ
Security	ก่อนที่จะโหลดตัวควบคุมของแอปพลิเคชัน (Application Controller) HTTP request และผู้ใช้ใดๆ ที่ส่งข้อมูลมาจะถูกกรองข้อมูลเพื่อความปลอดภัย
Application Controller	โหลดแบบจำลอง (Model), ไลบรารีหลัก (Libraries) , plugins, ผู้ช่วย (Helpers) และทรัพยากรอื่นๆ ที่จำเป็นในการทำงานที่ถูกร้องขอมา
View	ปฏิบัติงานและถูกส่งกลับไปยังเบราว์เซอร์เพื่อโชว์หน้าจอ ถ้าระบบแคชถูกใช้งาน หน้าจอจะถูกแคชก่อนแล้วจึงค่อยส่งสิ่งที่ร้องขอมาเป็นลำดับถัดไป

### 2.4.2 PHP

พีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพนซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ (ภาษาพีเอชพี, 2562) ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีการตอบโต้ได้อย่างรวดเร็ว ภาษาพีเอชพี เป็นส่วนประกอบภายในเว็บเพจ โดยคำสั่งจะปรากฏระหว่าง `<?php ... ?>` เช่น

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

```
<script language="php">
    echo "Hello World.";
</script>
```

ภาพที่ 2-3 คำสั่งการแสดงผลของ PHP

โครงสร้างควบคุมของ PHP จะมีความคล้ายคลึงกับ C/C++ มาก เช่น if , for , switch และมีส่วนที่คล้าย Perl สามารถกำหนดตัวแปรโดยไม่ต้อง กำหนดชนิดของตัวแปรว่าจะ เป็น int, float, boolean เป็นต้น

```
<?php
    for ($i = 0; $i < 10; $i++) {
        echo "Test".$i;
    }
?>
```

ภาพที่ 2-4 โครงสร้างควบคุมของ PHP

```

<?php
class MyClass
{
    private $var = 'สวัสดี PHPMy SAL localhost!';
    public function getHello()
    {
        return $this->var;
    }
}
$obj = new MyClass();
echo $obj->getHello();
?>

```

ภาพที่ 2-5 ตัวอย่างการเขียน php แบบ oop

การแสดงผลของพีเอชพี จะปรากฏในลักษณะ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถในการประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากดาต้าเบส ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะ CGI คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (command line scripting) ทำให้ผู้เขียนโปรแกรมสร้างสคริปต์พีเอชพี ทำงานผ่านพีเอชพี พาร์เซอร์ (PHP parser) โดยไม่ต้องผ่านเชิร์ฟเวอร์หรือเบราร์เซอร์ ซึ่งมีลักษณะเหมือนกับ Cron (ใน ยูนิกซ์หรือลินุกซ์) หรือ Task Scheduler (ในวินโดวส์) สคริปต์เหล่านี้สามารถนำไปใช้ในแบบ Simple text processing tasks ได้

การแสดงผลของพีเอชพี ถึงแม้ว่าจุดประสงค์หลักใช้ในการแสดงผล HTML แต่ยังสามารถสร้าง XHTML หรือ XML ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่างๆ ซึ่งสามารถแสดงผลข้อมูลหลัก PDF แฟลช (โดยใช้ libswf และ Ming) พีเอชพีมีความสามารถอย่างมากในการทำงานเป็นประมวลผลข้อความ จาก POSIX Extended หรือ รูปแบบ Perl ทั่วไป เพื่อแปลงเป็นเอกสาร XML ในการแปลงและ

เข้าสู่เอกสาร XML เรารองรับมาตรฐาน SAX และ DOM สามารถใช้รูปแบบ XSLT ของเราเพื่อแปลงเอกสาร XML

เมื่อใช้พีเอชพีในการทำอีคอมเมิร์ซ สามารถทำงานร่วมกับโปรแกรมอื่น เช่น Cybercash payment, CyberMUT, VeriSign Payflow Pro และ CCVS functions เพื่อใช้ในการสร้างโปรแกรมทำธุรกรรมทางการเงิน

คำสั่งของพีเอชพี สามารถสร้างผ่านทางโปรแกรมแก้ไขข้อความทั่วไป เช่น โน้ตแพด หรือ vi ซึ่งทำให้การทำงานของพีเอชพี สามารถทำงานได้ในระบบปฏิบัติการหลักเกือบทั้งหมด โดยเมื่อเขียนคำสั่งแล้วนำมาประมวลผล Apache, Microsoft Internet Information Services (IIS) , Personal Web Server, Netscape และ iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, และอื่นๆ อีกมากมาย. สำหรับส่วนหลักของ PHP ยังมี Module ในการรองรับ CGI มาตรฐาน ซึ่ง PHP สามารถทำงานเป็นตัวประมวลผล CGI ด้วย และด้วย PHP, คุณมีอิสรภาพในการเลือก ระบบปฏิบัติการ และ เว็บเซิร์ฟเวอร์ นอกจากนี้คุณยังสามารถใช้สร้างโปรแกรมโครงสร้าง สร้างโปรแกรมเชิงวัตถุ (OOP) หรือสร้างโปรแกรมที่รวมทั้งสองอย่างเข้าด้วยกัน แม้ว่าความสามารถของคำสั่ง OOP มาตรฐานในเวอร์ชันนี้ยังไม่สมบูรณ์ แต่ตัวไลบรารีทั้งหลายของโปรแกรม และตัวโปรแกรมประยุกต์ (รวมถึง PEAR library) ได้ถูกเขียนขึ้นโดยใช้รูปแบบการเขียนแบบ OOP เท่านั้น ซึ่งฐานข้อมูลส่วนหนึ่งที่รองรับได้แก่ ออราเคิล dBase PostgreSQL IBM DB2 MySQL Informix ODBC โครงสร้างของฐานข้อมูลแบบ DBX ซึ่งทำให้พีเอชพีใช้กับฐานข้อมูลอะไรก็ได้ที่รองรับรูปแบบนี้ และ PHP ยังรองรับ ODBC (Open Database Connection) ซึ่งเป็นมาตรฐานการเชื่อมต่อฐานข้อมูลที่ใช้กันแพร่หลายอีกด้วย คุณสามารถเชื่อมต่อกับฐานข้อมูลต่างๆ ที่รองรับมาตรฐานโลกนี้ได้

พีเอชพียังสามารถรองรับการสื่อสารกับการบริการในโพรโทคอลต่างๆ เช่น LDAP IMAP SNMP NNTP POP3 HTTP COM (บนวินโดวส์) และอื่นๆ อีกมากมาย คุณสามารถเปิด Socket บนเครือข่ายโดยตรง และ ตอบโต้โดยใช้ โพรโทคอลใดๆ ก็ได้ PHP มีการรองรับสำหรับการแลกเปลี่ยนข้อมูลแบบ WDDX Complex กับ Web Programming อื่นๆ ทั่วไปได้ พุดถึงในส่วน Interconnection, พีเอชพีมีการรองรับสำหรับ Java objects ให้เปลี่ยนมันเป็น PHP Object แล้วใช้งาน คุณยังสามารถใช้รูปแบบ CORBA เพื่อเข้าสู่ Remote Object ได้เช่นกัน

### 2.4.3 HTML

เอชทีเอ็มแอล (HTML: Hypertext Markup Language ภาษามาร์กอัปข้อความหลายมิติ) เป็นภาษามาร์กอัปหลักในปัจจุบันที่ใช้ในการสร้างเว็บเพจ หรือข้อมูลอื่นที่เรียกดูผ่านทางเว็บเบราว์เซอร์ ซึ่งตัวโค้ดจะแสดงโครงสร้างของข้อมูล ในการแสดง หัวข้อ ลิงก์ ย่อหน้า รายการ รวมถึงการสร้างแบบฟอร์ม เชื่อมโยงภาพหรือวิดีโอด้วย โครงสร้างของโค้ดเอชทีเอ็มแอลจะอยู่ในลักษณะภายในวงเล็บสามเหลี่ยม (จิราวุธ วารินทร์, 2560)

HTML ยังคงเป็นรูปแบบไฟล์อย่างหนึ่ง สำหรับ .html และ สำหรับ .htm ที่ใช้ในระบบปฏิบัติการที่รองรับ รูปแบบนามสกุล 3 ตัวอักษร

HTML เป็นภาษาหลักที่ใช้ในการเขียนเว็บเพจ โดยใช้ Tag ในการกำหนดการแสดงผล HTML ย่อมาจากคำว่า Hypertext Markup Language โดย Hypertext หมายถึง ข้อความที่เชื่อมต่อกันผ่านลิงค์ (Hyperlink) Markup language หมายถึงภาษาที่ใช้ Tag ในการกำหนดการแสดงผลสิ่งต่างๆ ที่แสดงอยู่บนเว็บเพจ ดังนั้น HTML จึงหมายถึง ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่ต่างก็เชื่อมถึงกันใน Hyperspace ผ่าน Hyperlink นั่นเอง (เอชทีเอ็มแอล, 2562)

ความเป็นมาของ HTML เริ่มขึ้นเมื่อปี 1980 เมื่อ Tim Berners Lee เสนอต้นแบบสำหรับนักวิจัยใน CERN เพื่อแลกเปลี่ยนเอกสาร ข้อมูลด้านการวิจัย โดยใช้ชื่อว่า Enquire ในปี 1990 ได้เขียนโปรแกรมเบราว์เซอร์ และทดลองรันบนเซิร์ฟเวอร์ ที่ถูกพัฒนาขึ้น HTML ได้รับการรู้จักจาก HTML Tag ซึ่งมีอยู่ 18 Tag ในปี 1991

HTML ถูกพัฒนาจาก SGML และ Tim ก็คิดเสมือนว่า HTML เป็นโปรแกรมน้อยของ SGML อยู่ในตอนนั้น ต่อมาในปี 1996 เพื่อกำหนดมาตรฐานให้ตรงกัน W3C World Wide Web Consortium จึงเป็นผู้กำหนดสเปกทั้งหมดของ HTML และปี 1999 HTML 4.01 ก็ถือกำเนิดขึ้น โดยมี HTML 5 ซึ่งเป็น Web Hypertext Application ถูกพัฒนาต่อมาในปี 2004 นอกจากนี้ยังมีการพัฒนาไปเป็น XHTML ซึ่งคือ Extended HTMLซึ่งมีความสามารถและมาตรฐานที่รัดกุมกว่าอีกด้วย โดยอยู่ภายใต้การควบคุมของ W3C (World Wide Web Consortium) (HTML คืออะไร, 2562)

### 2.4.3.1 ไทม์ไลน์ของ HTML

- ปี 1980 นักฟิสิกส์ Tim Berner-Lee เสนอระบบ ENQUIRE เพื่อแลกเปลี่ยนเอกสารใน CERN
- ปี 1989 Tim Berner-Lee เขียน Memo ถึง ระบบ Internet Based Hypertext System
- ปี 1990 Tim Berner-Lee เขียนเบราว์เซอร์และซอฟต์แวร์ฝั่งเซิร์ฟเวอร์
- ปี 1991 เอกสารเผยแพร่ HTML ชื่อ HTML Tag เผยแพร่ครั้งแรก มี Tag ทั้งหมด 18 Tag และมีการใช้งาน 11 Tag จวบจน HTML 4 กลางปี 1993 Tim และ Dan Conolly เสนอข้อกำหนดของ HTML เป็นครั้งแรกต่อสาธารณะ
- ปี 1994 HTML + HTML draft หมดอายุ HTML WORKING GROUP ดำเนินการต่อ
- ปี 1995 HTML 2.0 เสร็จสิ้น เป็นข้อกำหนด HTML แรกที่ถูกลำดับใช้เป็นมาตรฐานต่อไปในภายหน้า
- 24 พฤศจิกายน 1995: HTML 2.0 เผยแพร่เป็น RFC 1866 และบรรจุ RFC 2070 (สากล) 14 มกราคม 1997
- 12 กันยายน 1996: HTML 3.2 เผยแพร่เป็นคำแนะนำของ W3C เป็นรุ่นแรกที่ได้รับพัฒนาและได้รับมาตรฐานโดย W3C เนื่องจาก IETF ได้ปิดการทำงานของ HTML Working Group แล้ว
- 18 ธันวาคม 1997: HTML 4.0 ได้รับการเผยแพร่เป็นคำแนะนำ W3C ตอนแรกชื่อรหัสว่า "Cougar"
- พฤษภาคม 2000: HTML 4.01 เผยแพร่เป็นคำแนะนำ W3C มีรูปแบบ 3 รูปแบบเช่นเดียวกับ HTML 4.0
- 28 ตุลาคม 2014: HTML5 ได้รับการเผยแพร่เป็นคำแนะนำ W3C
- 1 พฤศจิกายน 2016: HTML 5.1 ได้รับการเผยแพร่เป็นคำแนะนำ W3C
- 14 ธันวาคม 2017: HTML 5.2 ได้รับการเผยแพร่เป็นคำแนะนำ W3C

### 2.4.3.2 โครงสร้าง HTML

#### ก) องค์ประกอบของภาษา HTML

เอกสาร html จะมีองค์ประกอบหลักๆ อยู่ 2 ส่วน คือ ส่วนที่ต้องการให้แสดงบนจอภาพและส่วนข้อความที่เป็นคำสั่ง โดยคำสั่งในเอกสาร html นี้จะเรียกว่า แท็ก (Tag) โดยแท็กจะต้องขึ้นต้นด้วย < ตามด้วย ชื่อแท็ก ปิดท้ายด้วย > ดังนี้ <Tag name> ซึ่งจะเรียกว่า แท็กเปิดแล้วจะต้องปิดท้ายข้อความด้วยแท็กปิด ซึ่งจะมีลักษณะดังนี้ </Tag name> ชื่อแท็กต่างๆ สามารถพิมพ์ตัวใหญ่หรือตัวเล็กก็ได้ความหมายเหมือนกัน

<Tag name> ข้อความที่ต้องการให้แสดง</Tag name>

### ภาพที่ 2-6 TAG HTML

#### ข) โครงสร้างของภาษา HTML

เอกสาร html จะมี โครงสร้างของภาษา html จะประกอบด้วย 2 ส่วนด้วยกัน คือ ส่วนที่เป็นส่วนหัว (head) และส่วนเนื้อหา (body) โดยมีรูปแบบภาษาดังนี้

```
<html>
  <head>
    <title>ข้อความที่ต้องการให้แสดงบนไต่เต้ลบาร์ </title>
  </head>
  <body>
    คำสั่งและข้อความให้แสดงบน browser
  </body>
</html>
```

### ภาพที่ 2-7 โครงสร้าง HTML

บล็อก `<html> ... </html>` เป็นบล็อกแรกที่จะต้องมีในเอกสาร และจะครอบคลุมบล็อกต่างๆ คือ เอกสาร html ทุกเอกสารจะต้องขึ้นต้นด้วย `<html>` และ ปิดท้ายด้วย `</html>` แต่ละ file และบล็อกอื่นๆ จะถูกเขียนอยู่ในบล็อกนี้ โดยจะมีบล็อกหลักๆ อยู่ 2 บล็อกก็คือ บล็อก head และ body

บล็อก `<head> ... </head>`

เป็นส่วนหัวเรื่องของเอกสาร ภายในจำเป็นจะมี บล็อก `<title>... </title>` ซึ่งจะเป็นแท็กผู้เขียน html นั้นใช้ตั้งเป็นไตเติ้ลสำหรับบอกโดยรวมว่าเอกสารนั้นต้องการเสนออะไร แล้วเวลาที่จะ bookmark ชื่อที่จะ save คือชื่อที่อยู่ใน บล็อก title นี้ ชื่อไตเติ้ลนี้จะต้องมีความยาวไม่เกิน 64 ตัวอักษร

บล็อก `<body> ... </body>` เป็นบล็อกที่บรรจุข้อมูลต่างๆ ที่ต้องการให้แสดงบน browser ไม่ว่าจะเป็น ข้อความ รูป ตาราง หรือแท็กที่ใช้ในการกำหนดรูปแบบของเอกสาร

comment (หมายเหตุ) หลังจากที่ทำกรเขียนโปรแกรมไปสักระยะ อาจจะนำโปรแกรมที่เขียนขึ้นมา นั้นไปพัฒนาต่อ ซึ่งอาจจะลืมเนื้อหาสาระสำคัญแล้วทำให้ดำเนินการเขียนต่อไม่ได้ จึงต้องมีการเขียน comment เพื่อช่วยเตือนความทรงจำได้

สำหรับคำสั่ง comment จะใช้ `<!--` เป็นแท็กเปิด และใช้ `-->` เป็นแท็กปิดข้อความที่อยู่ระหว่าง `<!-->` จะไม่ถูกแสดงบน เว็บเบราว์เซอร์ (โครงสร้าง HTML, 2562)

#### 2.4.4 CSS

CSS ย่อมาจาก Cascading Style Sheet มักเรียกโดยย่อว่า "สไตลชีต" คือภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS กำหนดกฎเกณฑ์ในการระบุรูปแบบ (หรือ "Style") ของเนื้อหาในเอกสาร อันได้แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งการกำหนดรูปแบบ หรือ Style นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสาร ไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลลัพท์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วกันทุกหน้าเอกสารภายในเว็บไซต์เดียวกัน โดยกฎเกณฑ์ในการกำหนดรูปแบบ (Style) เอกสาร HTML ถูกเพิ่มเข้ามาครั้งแรกใน HTML 4.0 เมื่อปีพ.ศ. 2539 ในรูปแบบของ CSS level 1 Recommendations ที่กำหนดโดย องค์กร World Wide Web Consortium หรือ W3C



### 2.4.1.1 ประโยชน์ของ CSS

ก) CSS มีคุณสมบัติมากกว่า tag ของ html เช่น การกำหนดกรอบให้ข้อความ รวมทั้งสี รูปแบบของข้อความที่กล่าวมาแล้ว

ข) CSS นั้นกำหนดที่ต้นของไฟล์ html หรือตำแหน่งอื่น ๆ ก็ได้ และสามารถมีผลกับเอกสารทั้งหมด หมายถึงกำหนด ครั้งเดียวจุดเดียวก็มีผลกับการแสดงผลทั้งหมด ทำให้เวลาแก้ไขหรือปรับปรุงทำได้สะดวก ไม่ต้องไล่ตามแก้ tag ต่าง ๆ ทั่วทั้งเอกสาร

ค) CSS สามารถกำหนดแยกไว้ต่างหากจาก ไฟล์เอกสาร html และสามารถนำมาใช้ร่วมกับเอกสารหลายไฟล์ได้ การแก้ไขก็แก้เพียง จุดเดียวก็มีผลกับเอกสารทั้งหมด

CSS กับ HTML / XHTML นั้นทำหน้าที่คนละอย่างกัน โดย HTML / XHTML จะทำหน้าที่ในการวางโครงสร้างเอกสารอย่างเป็นรูปแบบถูกต้อง เข้าใจง่าย ไม่เกี่ยวข้องกับการแสดงผล ส่วน CSS จะทำหน้าที่ในการตกแต่งเอกสารให้สวยงาม เรียกได้ว่า HTML /XHTML คือส่วน coding ส่วน CSS คือส่วน design (ดวงพร เกียรติคำ, 2560)

### 2.5.1.2 ตัวอย่างการจัดรูปแบบ HTML ด้วย CSS

```
<html>
  <body>
    <h1><font color="red" face="Arial">วิธีดูแลรักษาสุขภาพ</font></h1>
    <p>
      <font color="black" face="Arial">
        <b>
          รับประทานอาหารที่มีประโยชน์ หมั่นออกกำลังกาย และพักผ่อนให้เพียงพอ
        </b>
      </font>
    </p>
  </body>
</html>
```

ภาพที่ 2-8 ตัวอย่างกรณีการจัดรูปแบบการแสดงผลด้วยภาษา HTML

```

<html>
  <head>
    <style type="text/css">
      h1{color : red; font-family : Arial; }
      p {color : black; font-family : Arial; font-weight : bold}
    </style>
  </head>
  <body>
    <h1>วิธีดูแลรักษาสุขภาพ</h1>
    <p>รับประทานอาหารที่มีประโยชน์ หมั่นออกกำลังกาย และพักผ่อนให้เพียงพอ</p>
  </body>
</html>

```

ภาพที่ 2-9 การจัด css แทน code html

ด้านบนคือตัวอย่างเมื่อเปลี่ยนมาใช้คำสั่ง style sheet จัดรูปแบบการแสดงผลแทนการใช้ code ภาษา HTML ทำให้ code ภายในเอกสารอ่านเข้าใจง่าย และแก้ไขได้ง่ายขึ้น (CSS คืออะไร, 2562)

#### 2.4.5 JavaScript

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง Java JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) หรือเรียกว่า อ็อบเจ็กต์โอเรียนเตด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server) (รศ.ชาญชัย ศุภอรรถกร, 2561)

JavaScript ถูกพัฒนาขึ้นโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator2.0 เพื่อใช้สร้างเว็บเพจ

โดยติดต่อกับเวิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปจึงได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี 2538 แล้วตั้งชื่อใหม่ว่า JavaScript JavaScript สามารถทำให้ การสร้างเว็บเพจ มีลูกเล่น ต่าง ๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือ การกรอกข้อความในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ตาม สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ๆ ออกมาด้วย (ปัจจุบันคือรุ่น 1.5) ดังนั้น ถ้านำโค้ดของเวอร์ชันใหม่ ไปรันบนบราวเซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิด error ได้ (พื้นฐาน JavaScript เบื้องต้น, 2562)

#### 2.4.5.1 ความสามารถของ JavaScript

- ก) JavaScript ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายๆได้ โดยไม่ต้องพึ่งภาษาอื่น
- ข) JavaScript มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่นเมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox ก็สามารถสั่งให้เปิดหน้าต่างใหม่ได้ ทำให้เว็บไซต์ของเรามีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น นี่คือข้อดีของ JavaScript เลยก็ว่าได้ที่ทำให้เว็บไซต์ต่างๆทั้งหลายเช่น Google Map ต่างหันมาใช้
- ค) JavaScript สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้ นั่นคือสามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้ หรือหน้าแสดงเนื้อหาสามารถซ่อนหรือแสดงเนื้อหาได้แบบง่ายๆนั่นเอง
- ง) JavaScript สามารถใช้ตรวจสอบข้อมูลได้ สังเกตว่าเมื่อเรากรอกข้อมูลบางเว็บไซต์ เช่น Email เมื่อเรากรอกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่าเรากรอกผิด หรือลืมกรอกอะไรบางอย่าง เป็นต้น
- จ) JavaScript สามารถใช้ในการตรวจสอบผู้ใช้ได้เช่น ตรวจสอบว่าผู้ใช้ ใช้ web browser อะไร
- ฉ) JavaScript สร้าง Cookies (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

#### 2.4.5.2 ข้อดีและข้อเสียของ Java JavaScript

การทำงานของ JavaScript เกิดขึ้นบนเบราว์เซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้เซิร์ฟเวอร์อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (เรียกว่า server-side script) ดังนั้นจึงต้องใช้บนเซิร์ฟเวอร์ ที่สนับสนุนภาษาเหล่านี้เท่านั้น อย่างไรก็ตาม จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา server-side script อยู่ (ความจริง JavaScript ที่ทำงานบนเซิร์ฟเวอร์เวอร์ก็มี ซึ่งต้องอาศัยเซิร์ฟเวอร์ที่สนับสนุนโดยเฉพาะเช่นกัน แต่ไม่เป็นที่นิยมนัก) (JavaScript คืออะไร, 2562)

ในการใช้งาน JavaScript นั้น จำเป็นต้องใส่ Code ให้อยู่ระหว่างแท็ก <script> และ </script> โดยตัวคำสั่ง JavaScript นี้จะอยู่ในส่วนแท็ก <head> และ <body> ของเอกสาร HTML เช่น JavaScript ในแท็ก <body> ตัวอย่างเช่น

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h2>JavaScript in Body</h2>
6
7  <p id="demo"></p>
8
9  <script>
10 document.getElementById("demo").innerHTML = "My First JavaScript";
11 </script>
12
13 </body>
14 </html>
15

```

ภาพที่ 2-10 การใช้งาน JavaScript ในแท็ก <body>

### 2.4.5.3 รู้จักกับตัวแปรใน Javascript

**Global Variable** เป็นตัวแปรที่สามารถเรียกใช้ได้ทุกที่ นั่นคือประกาศไว้นอกฟังก์ชัน

**Local Variable** เป็นตัวแปรที่สามารถเรียกได้เฉพาะในฟังก์ชันใดฟังก์ชันหนึ่ง ที่มีการกำหนดตัวแปรไว้เท่านั้น นั่นคือได้ประกาศตัวแปรไว้ในฟังก์ชัน

### 2.4.5.4 การประกาศตัวแปร var let และ const ใช้งานต่างกันอย่างไร

ก) var เป็นการประกาศตัวแปรแบบ global scope คือ ประกาศค่าและใช้ค่าร่วมกันได้

```
var test = 123;
if (true) {
  var test = 456;
```

ภาพที่ 2-11 var การประกาศตัวแปรแบบ global scope

ข) let เป็นการประกาศตัวแปรแบบ block scope

```
var test = 123;
if (true) {
  var test = 456;
}
console.log(test); // 456
```

ภาพที่ 2-12 let การประกาศตัวแปรแบบ block scope

ค) Const เป็นการประกาศตัวแปรแบบ block scope ไม่สามารถเปลี่ยนแปลงค่าได้ แต่การเปลี่ยนค่าภายใน object

```
const PI = 3.14
PI = 1 // "PI" is read-only เปลี่ยนค่าไม่ได้อีกแล้วนะ
const obj = { a: 1 }
const obj = { a: 2 } // แต่การเปลี่ยนค่าภายใน object ไม่ได้ทำให้ memory
address เปลี่ยนไป obj.a = 2
console.log(obj) // { a: 2 }
```

ภาพที่ 2-13 Const เป็นการประกาศตัวแปรแบบ block scope

#### 2.4.5.5 Array ชุดของข้อมูล

เป็นตัวแปรพิเศษซึ่งสามารถเก็บมากกว่าหนึ่งค่าในแต่ละครั้งหากต้องการมีรายการสินค้า (ตัวอย่างเช่นชื่อรถ) การจัดเก็บรถยนต์ในตัวแปรเดียวอาจมีลักษณะดังนี้

```
var cars = ["Saab", "Volvo", "BMW"];
```

ภาพที่ 2-14 ตัวอย่างชุดอาเรย์ Cars

ในการเข้าถึงของอาร์เรย์ค่าเริ่มต้นของ Index จะมีค่าเท่ากับ 0 เสมอ

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h2>JavaScript Arrays</h2>
6
7 <p>JavaScript array elements are accessed using numeric indexes (starting from 0).</p>
8
9 <p id="demo"></p>
10
11 <script>
12 var cars = ["Saab", "Volvo", "BMW"];
13 document.getElementById("demo").innerHTML = cars[0];
14 </script>
15
16 </body>
17 </html>
18
19

```

ภาพที่ 2-15 ตัวอย่างชุดอาร์เรย์ Cars

#### 2.4.5.6 function () ฟังก์ชัน

ฟังก์ชัน หรือชุดคำสั่งที่รวม Statement การทำงานเอาไว้ สามารถเรียกชื่อมาใช้งานตามที่ต้องการใช้ได้

```

function hello () {
    .....
}
Hello (); // เรียกใช้งาน

```

ภาพที่ 2-16 ตัวอย่างการเรียกใช้ฟังก์ชัน เมื่อสร้างไว้เบื้องต้น

#### 2.4.5.7 Loop ใช้ทำซ้ำตามเงื่อนไขที่กำหนด Loop for

```
for (ตัวแปรเริ่มต้น; เงื่อนไข (Condition); เพิ่มค่าตัวแปร ) {
    Statement ที่ต้องการให้ทำงาน ถ้าเงื่อนไขยังเป็นจริงอยู่
}
```

ภาพที่ 2-17 Loop ใช้ทำซ้ำตามเงื่อนไขที่กำหนด Loop for

```
for (var i = 1; i < 5; i++) {
    console.log(i);
} /* ผลลัพธ์เราจะได้ 1 2 3 4 เพราะรอบสุดท้าย i มีค่าเป็น 5 ซึ่งจะทำให้ส่วนเงื่อนไขเป็นเท็จ เป็นการ
หยุด loop */
```

ภาพที่ 2-18 ตัวอย่างการทำงานของ Loop for

#### 2.4.6 XAMPP

Xampp เป็นโปรแกรม Apache web server ไว้จำลอง web server เพื่อไว้ทดสอบ สคริปต์ หรือเว็บไซต์ในเครื่อง โดยที่ไม่ต้องเชื่อมต่ออินเทอร์เน็ตและไม่ต้องมีค่าใช้จ่ายใดๆ ง่ายต่อการติดตั้งและใช้งานโปรแกรม Xampp จะมาพร้อมกับ PHP ภาษาสำหรับพัฒนาเว็บแอปพลิเคชันที่เป็นที่นิยม, MySQL ฐานข้อมูล, Apache จะทำหน้าที่เป็นเว็บ เซิร์ฟเวอร์, Perl อีกทั้งยังมาพร้อมกับ OpenSSL , phpMyAdmin (Xampp คืออะไร, 2562)

ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล สนับสนุนฐานข้อมูล MySQL และ SQLite โปรแกรม Xampp จะอยู่ในรูปแบบของไฟล์ Zip, tar, 7z หรือ exe โปรแกรม Xampp อยู่ภายใต้ใบอนุญาตของ GNU General Public License แต่บางครั้งอาจจะมีการเปลี่ยนแปลงเรื่องของลิขสิทธิ์ในการใช้งาน จึงควรติดตามและตรวจสอบโปรแกรมด้วย



#### 2.4.6.1 โปรแกรม XAMPP สามารถใช้งานได้ 4 OS ได้แก่

- Windows ใช้งานได้กับ windows รุ่น 2000, 2003, xp, vista, windows 7
- Linux สำหรับ SuSE, RedHat, Mandrake, Debian และ Ubuntu
- Mac OS X
- Solaris สำหรับ Solaris 8 และ Solaris 9

ในการใช้งานเบื้องต้นให้ดับเบิลคลิก Xampp Control Panel Application แล้วทำการคลิกปุ่ม start จากนั้นสามารถใช้งานได้ โดยเปิด Browser ขึ้นมาพิมพ์ localhost หรือ 127.0.0.1

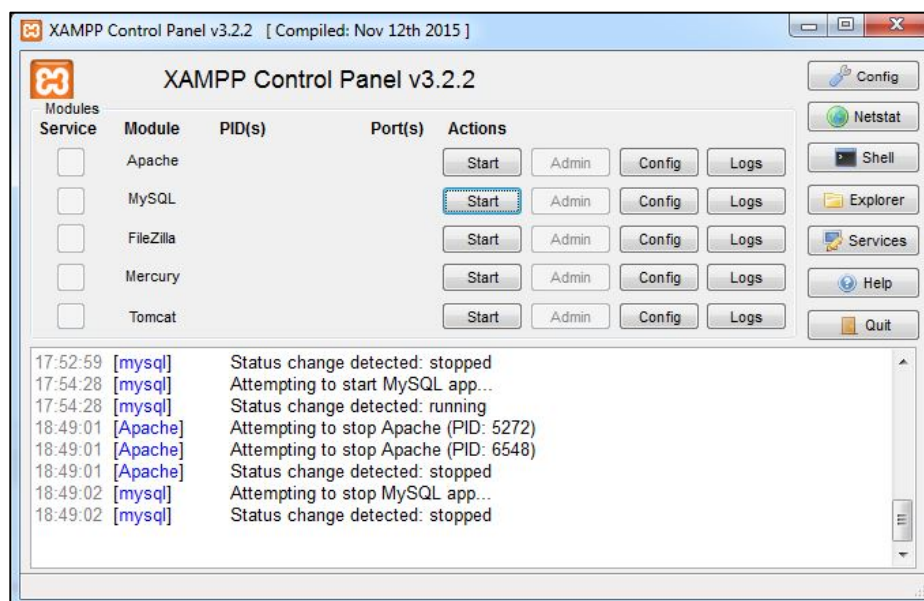
#### 2.4.6.2 ข้อกำหนดด้านเทคนิค

- เครื่องคอมพิวเตอร์ควรมี RAM ไม่ต่ำกว่า 128 MB
- Harddisk มีพื้นที่มากกว่า 320 MB
- CPU ไม่กำหนดขั้นต่ำ

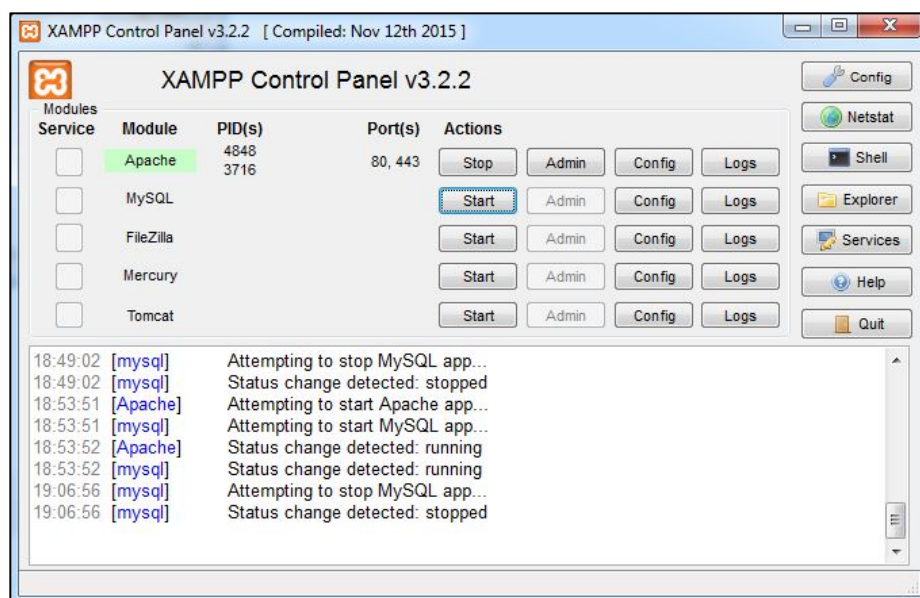
จะเห็นได้ว่า Xampp เป็นโปรแกรม Apache web server ที่ได้รับความนิยมมากในปัจจุบันและยังมีการพัฒนาอย่างต่อเนื่อง โดยเวอร์ชันล่าสุดของ Xampp คือ PHP 7.3.9 (Xampp คืออะไร, 2562)

#### 2.4.6.3 ขั้นตอนการเปิดใช้งานเซิร์ฟเวอร์

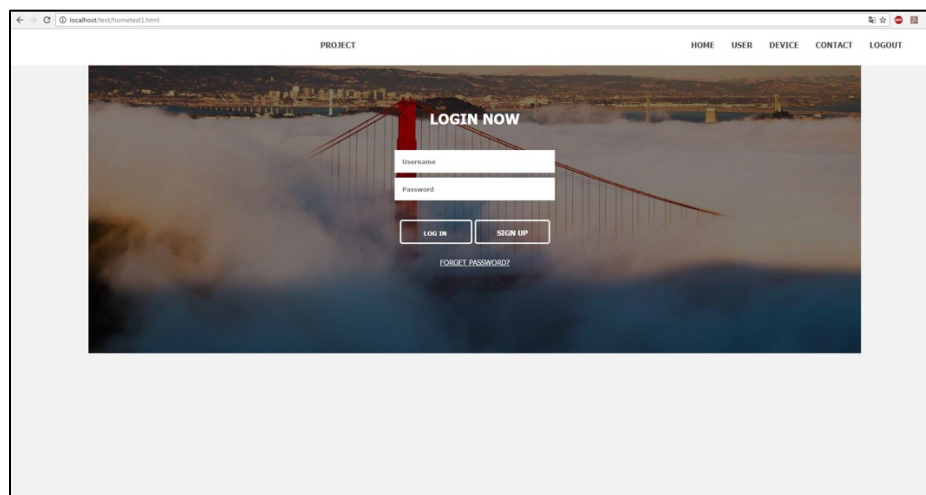
การเปิดใช้งาน Xampp เพื่อจำลองเซิร์ฟเวอร์ สำหรับการจำลองเว็บไซต์เสมือนการใช้งานจริงบนเซิร์ฟเวอร์ ซึ่งสามารถดาวน์โหลด Xampp ตัวติดตั้งได้ที่ เว็บไซต์ [apachefriends.org](https://www.apachefriends.org) หรือเข้าไปที่ URL : <https://www.apachefriends.org/index.html> เมื่อติดตั้งแล้วสามารถทำตามภาพประกอบด้านล่างเพื่อเปิดใช้งาน



ภาพที่ 2-19 เปิดโปรแกรม Xampp

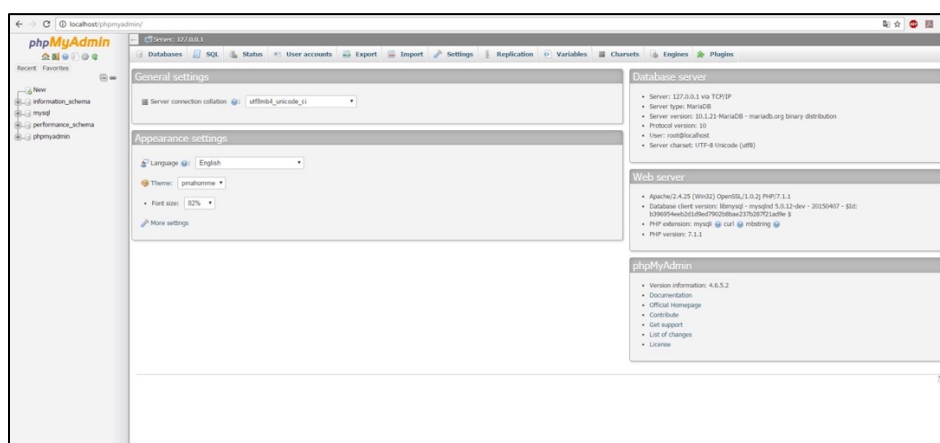


ภาพที่ 2-20 กดปุ่ม start บน module Apache



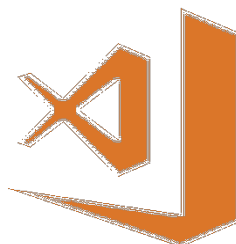
ภาพที่ 2-21 เปิด website ที่ต้องการจำลอง server

เมื่อจำลองเซิร์ฟเวอร์ได้แล้ว สามารถเข้าไปที่ <http://localhost/phpmyadmin/> เพื่อเข้าไปสร้างฐานข้อมูล



ภาพที่ 2-22 phpMyAdmin สร้างฐานข้อมูล

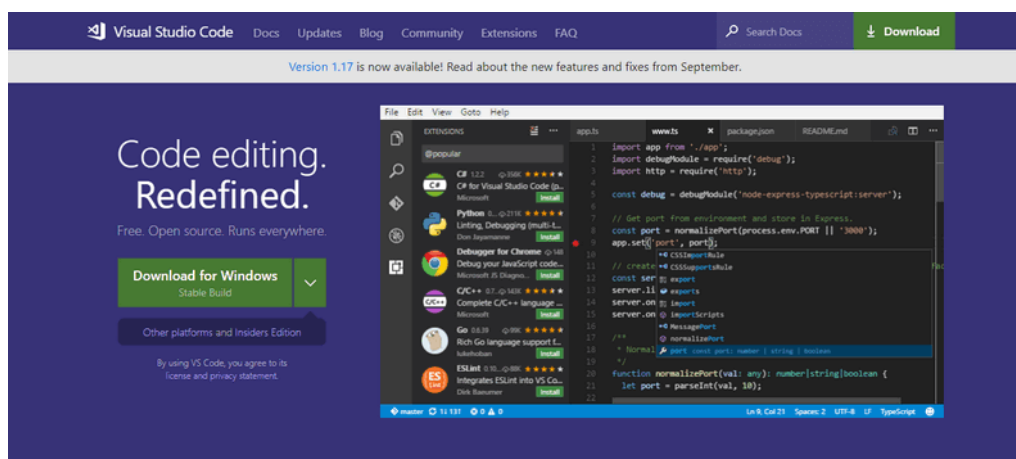
### 2.4.7 Visual Studio Code



ภาพที่ 2.23 Visual Studio Code

Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพ (Visual Studio Code, 2562)

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



ภาพที่ 2-24 Code editing

ความแตกต่างระหว่าง VSCode และ Visual Studio คือ

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

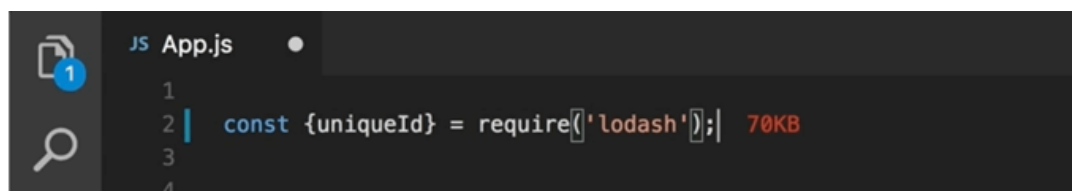
สำหรับผู้สนใจใช้งาน สามารถดาวน์โหลดได้ที่ <https://code.visualstudio.com/>

#### 2.4.7.1 Visual Studio Code Extensions

(10 Visual Studio Code Extensions, 2562) ตัว Visual Studio Code ยังมีตัวช่วยเสริมเพื่อให้ นักพัฒนาสามารถใช้งานได้ง่ายขึ้น โดยจะยกตัวอย่าง Extensions ให้พออนุภาพโดยรวมดังนี้

##### ก) Import Cost

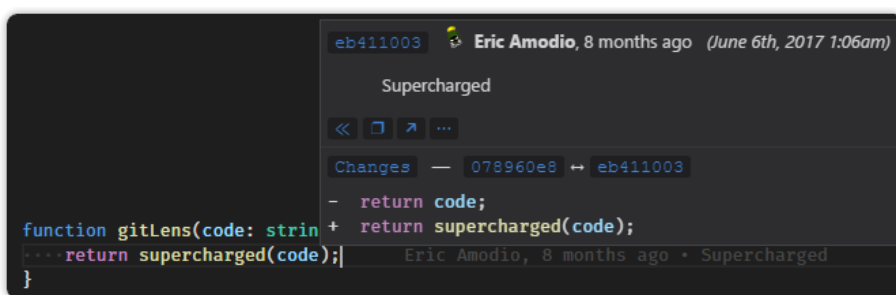
Import cost เป็น extension ที่จะแสดงขนาดไฟล์ของ package ที่ import เข้ามา โดยจะแสดงให้เห็นทันทีว่า package ตัวไหนมีขนาดไฟล์ที่มากเกินไป



ภาพที่ 2-25 Import Cost

### ข) GitLens — Git supercharged

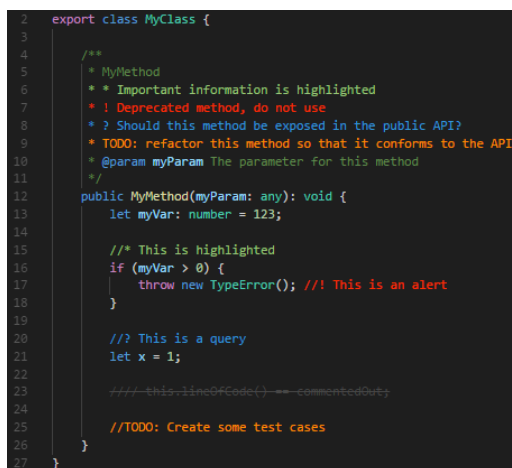
ถึงแม้ว่า Visual Studio Code จะมีการฝัง Git เข้ามาในตัว editor อยู่แล้ว แต่ก็สามารถที่จะเพิ่มพลังให้มันได้อีกด้วย GitLens โดยเมื่อเอา cursor ไปวางไว้ที่โค้ด ก็จะได้เห็นได้ว่าใครเป็นคน commit โค้ดส่วนนั้นขึ้นไป เมื่อไหร่ แล้วยังดู changes ย้อนหลังได้อีกด้วย



ภาพที่ 2-26 GitLens — Git supercharged

### ค) Better Comments

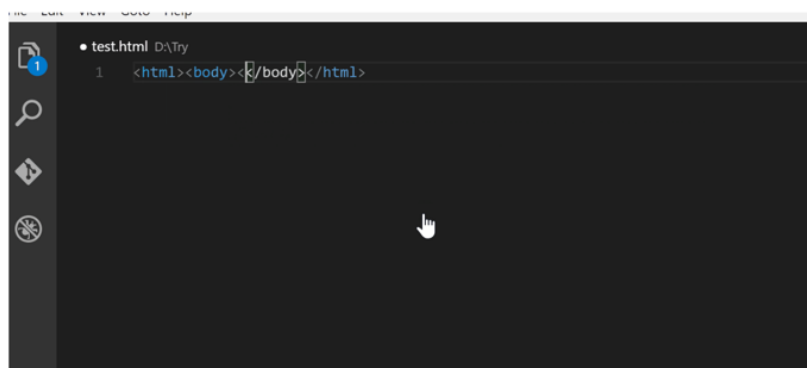
บางทีการเขียน TODO ก็อาจจะกลืนไปกับ comment ทั่วไปมากเกินไปจนแยกไม่ออกว่าอันไหนคือ comment อันไหนคือ TODO แต่ Better Comments จะช่วยแยกสีให้ชัดเจน



ภาพที่ 2-27 Better Comments

### ง) Auto Close Tag

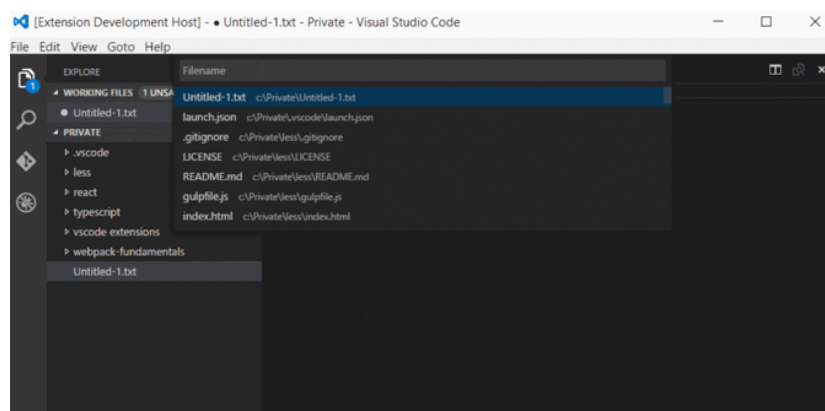
เวลาเขียนโค้ดบางทีก็อยากทำงานเร็วๆ เขียนแค่ tag เปิดก็พอ ซึ่ก็เขียน tag ปิด ควรที่จะลง Auto Close Tag ไว้ให้ editor ทำการปิด tag ให้อัตโนมัติ



ภาพที่ 2-28 Auto Close Tag

### จ) Relative Path

การเขียนโปรเจกใหญ่ๆ ที่มี folders และ files เยอะๆ จะทำให้พบกับปัญหาการ import file ให้ถูก path Relative Path จะช่วย import file ให้แบบสะดวก รวดเร็ว ถูกต้อง แม่นยำ



ภาพที่ 2-29 Relative Path

### ฉ) Auto Rename Tag

ใครที่เขียน React หรือ Vue บ่อยๆ ก็มักจะต้องคอยเปลี่ยนชื่อ component tag เปิด และ ปิด ให้ตรงกันอยู่เสมอ ซึ่งก็มีตัวช่วยอย่าง Auto Rename Tag ที่จะคอยจับคู่ tag เปิด/ปิด ทำให้สามารถแก้ไขชื่อ tag แค่นี้เดียว



ภาพที่ 2-30 Auto Rename Tag

### ช) npm intellisense

ใครที่เขียน JavaScript ก็คงจะต้องมีการเรียกใช้งาน npm packages โดยการ import/require npm package แต่ละทีก็ต้อง import/require ชื่อ package ให้ถูก ซึ่งตัว npm intellisense จะช่วย autocomplete ชื่อ package ให้ ทำให้สามารถทำงานต่อได้อย่างรวดเร็ว



ภาพที่ 2-31 npm intellisense



### ซ) Bracket Pair Colorizer

หลายๆ คนมักจะเจอปัญหาว่า ลืมใส่วงเล็บ หรือมีปีกกาเกินมา อยู่เป็นประจำ แถมนางที่ก็ดูไม่ค่อยออกกว่าอะไรจับคู่กับอะไรอยู่ แต่ Bracket Pair Colorizer จะช่วยให้มองภาพออกได้เป็นแบบสีรุ้ง

```

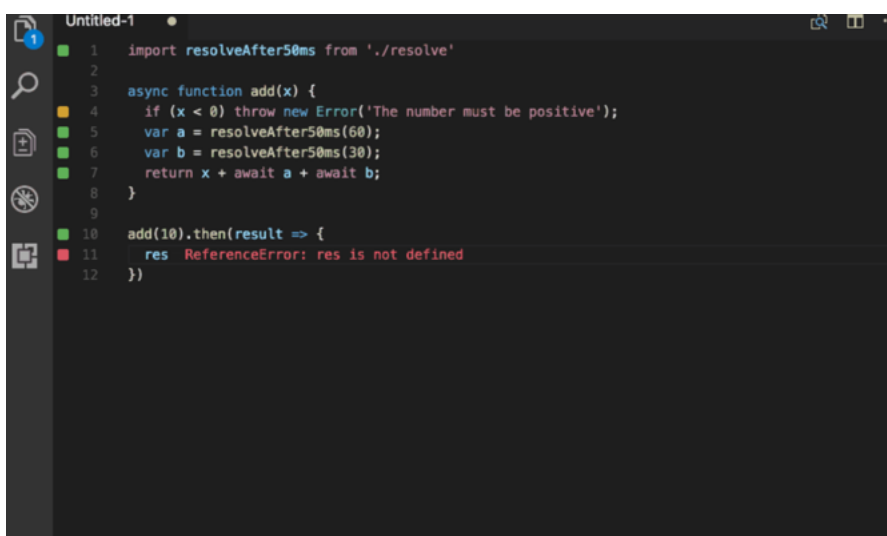
22  public updateDocument(document: TextDocument) {
23      const documentDecoration = this.getDocumentDecorations(document);
24      if (documentDecoration) {
25          documentDecoration.triggerUpdateDecorations();
26      }
27  }

```

ภาพที่ 2-32 Bracket Pair Colorizer

### ฎ) Quokka.js

Quokka.js เป็นของเล่นที่จะทำให้เหล่านักพัฒนา JavaScript ทำการเขียนโค้ดได้สนุกมากยิ่งขึ้น เพราะสามารถที่จะดูผลลัพธ์ได้ทันทีว่า function นี้ return ค่าอะไรออกมา หรือจะดูค่าของ array หรือ object ก็ยังได้ แถมนยังมีการเช็ค performance ของ function แบบ real-time ได้อีกด้วย



```

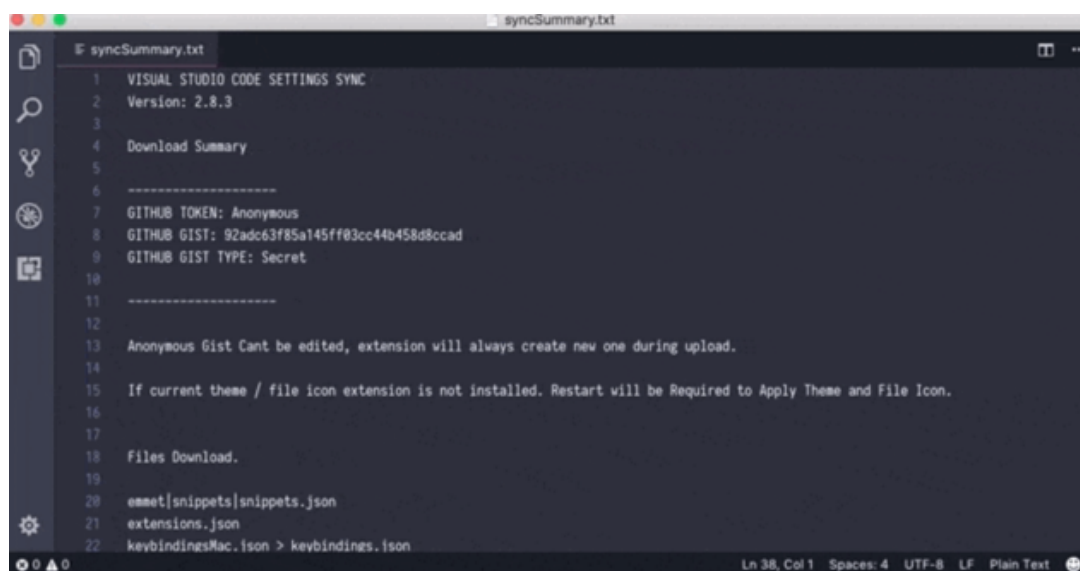
1  import resolveAfter50ms from './resolve'
2
3  async function add(x) {
4      if (x < 0) throw new Error('The number must be positive');
5      var a = resolveAfter50ms(60);
6      var b = resolveAfter50ms(30);
7      return x + await a + await b;
8  }
9
10 add(10).then(result => {
11     res  ReferenceError: res is not defined
12 })

```

ภาพที่ 2-33 Quokka.js

### ก) Settings Sync

คนที่ทำงานในคอมหลายๆ เครื่อง มักจะต้องเสียเวลาคอยตั้งค่าให้ Visual Studio Code นั้นมีหน้าตาและการตั้งค่าที่เหมือนกัน แต่จะง่ายกว่ามั้ยถ้าตั้งค่าที่เครื่องเดียว แล้วให้อีกเครื่องนั้นเปลี่ยนตามไปด้วย? Settings Sync จะช่วยอัปโหลดการตั้งค่าต่างๆ ขึ้น GitHub ทำให้สามารถไปดาวน์โหลดการตั้งค่าไปใช้ในคอมเครื่องอื่นๆ ได้อย่างสะดวกสบายนั่นเอง



```

1  VISUAL STUDIO CODE SETTINGS SYNC
2  Version: 2.8.3
3
4  Download Summary
5
6  -----
7  GITHUB TOKEN: Anonymous
8  GITHUB GIST: 92adc63f85a145ff03cc44b458d8ccad
9  GITHUB GIST TYPE: Secret
10
11 -----
12
13 Anonymous Gist Cant be edited, extension will always create new one during upload.
14
15 If current theme / file icon extension is not installed. Restart will be Required to Apply Theme and File Icon.
16
17
18 Files Download.
19
20 emmet|snippets|snippets.json
21 extensions.json
22 keybindingsMac.json > keybindings.json
  
```

ภาพที่ 2-34 Settings Sync

## 2.5 งานวิจัยที่เกี่ยวข้อง

วีรชน นามโคตร (2553). การพัฒนาระบบมีการจัดเก็บบันทึกข้อมูลของนักเรียน ข้อมูลวิชาเรียน ข้อมูลผู้สอน ข้อมูลสุขภาพ ข้อมูลประกันชีวิต ข้อมูลสมาชิกห้องสมุด ข้อมูลหนังสือ การพัฒนาระบบตรวจสอบการเข้าเรียนให้นักเรียนแสดงบัตรประจำตัวผ่านเครื่องตรวจสอบขณะที่เดินเข้าห้องเรียน ซึ่งเครื่องตรวจบัตรจะทำการอ่านบาร์โค้ดที่อยู่บนบัตรประจำตัวนักเรียนผ่านเครื่องอ่านบาร์โค้ด แบบมือถือ แล้วนำข้อมูลที่ได้อาตรวจสอบกับรหัสที่ฐานข้อมูลหลักจนกว่าจะสามารถระบุตัว บุคคลได้และทำการเก็บข้อมูลจึงทำการเก็บข้อมูล การพัฒนาระบบการเข้ารับบริการพยาบาลให้ นักเรียนแสดงบัตรประจำตัวผ่านเครื่องอ่านบาร์โค้ดตรวจสอบข้อมูลประวัติทางสุขภาพของนักเรียน ทำการกรอกข้อมูลของผู้เข้ารับบริการ จากนั้นทำการบันทึกข้อมูลไปยังฐานข้อมูลกลาง การเข้ารับ บริการยืม-คืน หนังสือระบบจะตรวจสอบข้อมูลจากบัตรประจำตัวนักเรียนว่าอยู่ในระบบสมาชิก หรือไม่ ถ้าอยู่ในระบบสมาชิกและมีสิทธิในการยืม หนังสือระบบจะทำการอนุญาตให้ยืมหนังสือได้ ผลที่ได้จากการวิจัยแสดงให้เห็นว่า การนำบัตรประจำตัวนักเรียนแบบบาร์โค้ดมาใช้ แทนระบบเดิมนั้นทำให้การทำงานของครูผู้สอนและเจ้าหน้าที่ผู้ปฏิบัติด้านต่างของโรงเรียนวัดอินทารามมีความสะดวกรวดเร็วยิ่งขึ้น ผลจากการประเมินความพึงพอใจของผู้ทดสอบจากผู้ใช้ 60 คน พบว่าในด้านการใช้งานของระบบผู้ใช้มีความพึงพอใจโดย โดยมีค่าเฉลี่ยเท่ากับ 3.7 อยู่ในระดับที่ดี ดังนั้นผลการวิจัยในครั้งนี้สามารถนำไปประยุกต์ใช้ในระบบสารสนเทศของโรงเรียนวัดอินทาราม ให้เกิดประสิทธิภาพในการทำงานมากยิ่งขึ้น

ศิวพร เพชรวิสัยและนันทนา ศรีพรมทอง (2561). ระบบการจัดการอบรมและจองห้องสัมมนาออนไลน์ในรูปแบบของ เว็บแอปพลิเคชัน โดยรองรับการทำงานในลักษณะออนไลน์เพื่อช่วยอำนวยความสะดวกในการจัดการอบรมและจองห้องสัมมนา ของสำนักวิทยบริการและเทคโนโลยีสารสนเทศ มีการจัดทำหลักสูตรและการออกแบบใบประกาศนียบัตร ทำให้มีความปลอดภัยในการเก็บและสำรองข้อมูลได้ง่าย เพราะเนื่องจากการจัดเก็บที่ไม่เป็นระบบอาจเกิดการสูญหายของข้อมูลต่าง ๆ ทำให้ไม่สามารถค้นหาข้อมูลที่ต้องการได้และข้อมูลมีความซ้ำซ้อน ดังนั้นเพื่อให้โครงการนี้สามารถแก้ไขปัญหาที่ต้องการได้ จึงได้ออกแบบและจัดทำโครงการนี้ขึ้นมาเพื่อที่จะให้การทำงานของเจ้าหน้าที่ของสำนักวิทยบริการและเทคโนโลยีสารสนเทศ สามารถจัดการอบรมและจองห้องสัมมนาได้อย่างสะดวกและรวดเร็ว รวมถึงผู้ใช้บริการได้รับความสะดวกมากขึ้น

## 2.6 การประเมินความพึงพอใจ

### 2.6.1 การสร้างแบบประเมินความพึงพอใจของระบบ

หลังจากการพัฒนากระบวนการบริหารจัดการร้านค้า ผู้จัดทำได้ทำการเพิ่มข้อมูลสินค้า รวมถึงข้อมูลอื่นที่เกี่ยวข้องเพื่อทดสอบระบบด้วยแบบประเมินความพึงพอใจ

2.6.1.1 สร้างแบบประเมินความพึงพอใจให้สอดคล้องกับรายละเอียดทั้ง 3 ด้าน คือ ด้านเนื้อหา ด้านการออกแบบ ด้านการนำไปใช้งาน โดยภาพรวมของระบบมีการใช้แบบมาตราส่วนประเมินค่า (Rating Scale) กำหนดการให้คะแนนความพึงพอใจเป็น 5 ระดับ โดยพิจารณาจากคะแนนเฉลี่ยของผู้ใช้งานระบบต้องมีคะแนนเฉลี่ยอยู่ในระดับดีหรือตั้งแต่ 3.51 คะแนนขึ้นไปหรือมีความพึงพอใจอยู่ในระดับมาก จึงจะยอมรับว่าระบบมีประสิทธิภาพในการใช้งานได้ ซึ่งเกณฑ์การให้คะแนนของแบบประเมินความพึงพอใจดังแสดงในตารางที่ 2-5 และเกณฑ์การแปลผลคะแนนของแบบประเมินความพึงพอใจ ดังแสดงในตารางที่ 2-6

ตารางที่ 2-5 เกณฑ์การให้คะแนนของแบบประเมินความพึงพอใจ

ระดับเกณฑ์การให้คะแนน	ความหมาย
5	มีความพึงพอใจต่อระบบมากที่สุด
4	มีความพึงพอใจต่อระบบมาก
3	มีความพึงพอใจต่อระบบปานกลาง
2	มีความพึงพอใจต่อระบบน้อย
1	มีความพึงพอใจต่อระบบน้อยมาก

ตารางที่ 2-6 เกณฑ์การแปลผลคะแนนของแบบประเมินความพึงพอใจ

ระดับคะแนน		ความหมาย
เชิงคุณภาพ	เชิงปริมาณ	
ดีมาก	4.51 - 5.00	มีความพึงพอใจอยู่ในระดับมากที่สุด
ดี	3.51 - 4.50	มีความพึงพอใจอยู่ในระดับมาก
ปานกลาง	2.51 - 3.50	มีความพึงพอใจอยู่ในระดับปานกลาง
น้อย	1.51 - 2.50	มีความพึงพอใจอยู่ในระดับน้อย
ควรปรับปรุง	1.00 - 1.50	มีความพึงพอใจอยู่ในระดับน้อยที่สุด

การหาค่าคะแนนเฉลี่ย (Mean)

$$\bar{x} = \frac{\sum x}{n}$$

เมื่อ  $\bar{x}$  คือ ค่าคะแนนเฉลี่ย  
 $\sum x$  คือ ผลรวมของคะแนนทั้งหมด  
 $n$  คือ จำนวนคน

การหาค่าส่วนเบี่ยงเบนมาตรฐาน :S.D.

$$s.d. = \sqrt{\frac{n \sum x^2 - (\sum x)^2}{n(n-1)}}$$

เมื่อ  $s.d.$  คือ ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation)  
 $\sum x^2$  คือ ผลบวกของกำลังสองของคะแนนแต่ละตัว  
 $\sum x$  คือ ผลบวกของคะแนนแต่ละตัว  
 $n$  คือ จำนวนคน