

## บทที่ 2

### เอกสารและงานวิจัยที่เกี่ยวข้อง

ผู้จัดทำได้ศึกษารายละเอียดของเอกสารและงานวิจัยที่เกี่ยวข้อง เพื่อจัดทำ “ระบบจองคิวสอบ โพรเจกออนไลน์” ดังนี้

- 2.1 CodeIgniter (ภาษาที่ใช้เขียนระบบ)
- 2.2 PHP (ภาษาที่ใช้เขียนระบบ)
- 2.3 HTML (ภาษาที่ใช้เขียนระบบ)
- 2.4 CSS (ภาษาที่ใช้เขียนระบบ)
- 2.5 JavaScript (ภาษาสคริปต์เชิงวัตถุ)
- 2.6 XAMPP (โปรแกรมจำลองเว็บเซิร์ฟเวอร์)
- 2.7 Visual Studio Code (โปรแกรม Code Editor)
- 2.8 การประเมินความพึงพอใจ

#### 2.1 CodeIgniter

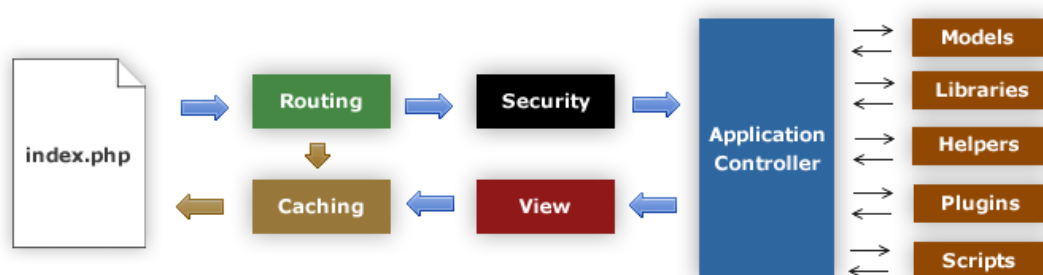
CodeIgniter [1] คือ framework ที่ถูกพัฒนาขึ้นด้วย ภาษา PHP สำหรับ CodeIgniter Framework เป็นเครื่องมือสำหรับช่วยพัฒนาเว็บไซต์และเว็บแอปพลิเคชัน โดยมีโครงสร้างการพัฒนาโปรแกรมอย่างเป็นระบบและรวมคำสั่งต่างๆ ที่จำเป็นต่อการพัฒนาเว็บไซต์ด้วย PHP ไว้ภายใน CodeIgniter พัฒนาโดย Rick Ellis ซึ่งมีลิขสิทธิ์เป็น Open Source สามารถดาวน์โหลดและใช้ได้ฟรี ถูกพัฒนาขึ้นตั้งแต่ปี 2006 และยังมีการพัฒนามาเรื่อยๆ จนถึงปัจจุบัน

##### 2.1.1 โครงสร้างของ CodeIgniter Structure

CodeIgniter รองรับการทำงานแบบ 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 เพียงไม่กี่ที่เท่านั้น ส่วนระบบ url ของระบบ codeigniter สามารถสร้างได้สะอาดและเป็นมิตรกับกลไกการค้นหา (Search-Engine) ทำให้สามารถนำไปพัฒนาได้ทั้ง เว็บไซต์ และเว็บแอปพลิเคชัน ไฟล์ภายใน CodeIgniter ประกอบด้วย โครงสร้างหลักๆ คือ application, system, index.php ระบบภายในสามารถแยกโปรเจกออกเป็นหลายๆ application ได้

โครงสร้างไฟล์ภายในโฟลเดอร์ application ที่ต้องใช้ มีดังต่อไปนี้

cache	เป็นส่วนเก็บข้อมูลที่ประมวลผลแล้ว เพื่อการเรียกใช้ครั้งต่อไป
config	ใช้ตั้งค่าระบบต่างๆ ภายใน application
controllers	เป็นส่วนรวมโค้ดการประมวลผล จะเขียนโค้ด PHP เป็นส่วนใหญ่
core	เป็นส่วนเสริมที่มีไว้สำหรับสร้างการใช้งาน (MVC) ขึ้นอีกชั้นหนึ่ง ในชื่อ modules
helpers	หากต้องการเขียนฟังก์ชันใช้เองในระบบสามารถสร้างและเก็บไว้ในที่นี่
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 ที่นี้ แต่นามสกุลไฟล์ต้องเป็น .php เท่านั้น



ภาพที่ 2.1 Model-View-Controller MVC

Model-View-Controller MVC คือโครงสร้างตามหลักการ ระบบจะแยกส่วนประมวลผล ออกจาก ส่วนแสดงผล แต่ในทางปฏิบัติ สามารถเขียนโค้ด PHP ลงในส่วนแสดงผลได้เช่นกัน

Models : อธิบายโครงสร้างข้อมูล แบบฉบับของคลาสแบบจำลองจะมีฟังก์ชันที่ช่วยให้รับ , ใส่ , ปรับปรุงข้อมูลในฐานข้อมูล

View : คือข้อมูลที่ถูกแสดงให้กับผู้ใช้ View ส่วนใหญ่จะเป็นเว็บเพจ แต่ใน CodeIgniter , view สามารถเป็นเพียงบางส่วนของหน้าได้ เช่น ส่วนหัว ส่วนท้าย มันสามารถจะเป็นหน้า RSS หรือเป็นอย่างอื่นได้ที่เป็น "เพจ"

Controller : ถูกใช้เป็น สื่อกลาง ระหว่าง Model , View และแหล่งที่มาอื่นๆ ที่ต้องการในการทำงาน HTTP request และการสร้างเว็บเพจ

### 2.1.2 ขั้นตอนการทำงาน

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

## 2.2 PHP

พีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพนซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีการตอบโต้ได้อย่างรวดเร็ว

ภาษาพีเอชพี เป็นส่วนประกอบภายในเว็บเพจ โดยคำสั่งจะปรากฏระหว่าง `<?php ... ?>` เช่น

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

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

โครงสร้าง ควบคุมของ PHP จะมีความคล้ายคลึงกับ C/C++ มาก เช่น if , for , switch และมีบางส่วนที่คล้าย Perl สามารถกำหนดตัวแปรโดยไม่ต้อง กำหนดชนิดของตัวแปรว่าจะเป็น int, float, boolean เป็นต้น

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

ตัวอย่างการเขียน php แบบ oop

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

$obj = new MyClass();
echo $obj->getHello();

?>
```

การแสดงผลของพีเอชพี จะปรากฏในลักษณะ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถในการประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากดาต้าเบส ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะCGI คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (command line scripting) ทำให้ผู้เขียนโปรแกรมสร้างสคริปต์พีเอชพี ทำงานผ่านพีเอชพี พาร์เซอร์ (PHP parser) โดยไม่ต้องผ่านเชิร์ฟเวอร์หรือเบราวเซอร์ 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

ในการแปลงและเข้าสู่เอกสาร 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 นอกจากนี้คุณยังสามารถใช้สร้างโปรแกรมโครงสร้างสร้างโปรแกรมเชิงวัตถุ (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, พีเอชพี มีการรองรับสำหรับ Java objects ให้เปลี่ยนมันเป็น PHP Object แล้วใช้งาน คุณยังสามารถใช้รูปแบบ CORBA เพื่อเข้าสู่ Remote Object ได้เช่นกัน

## 2.3 HTML

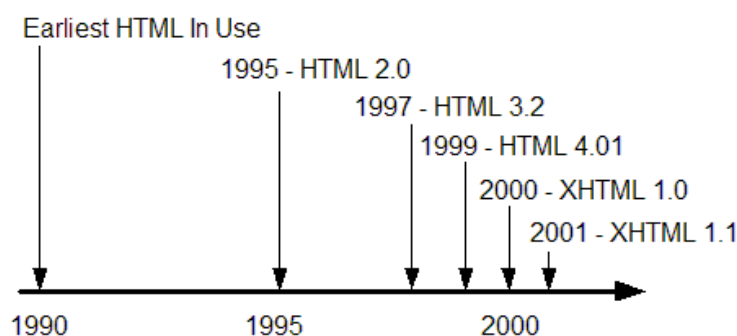
เอชทีเอ็มแอล (HTML: Hypertext Markup Language ภาษามาร์กอัปข้อความหลายมิติ) เป็นภาษามาร์กอัปหลักในปัจจุบันที่ใช้ในการสร้างเว็บเพจ หรือข้อมูลอื่นที่เรียกดูผ่านทางเว็บเบราว์เซอร์ ซึ่งตัวโค้ดจะแสดงโครงสร้างของข้อมูล ในการแสดง หัวข้อ ลิงก์ ย่อหน้า รายการ รวมถึงการสร้างแบบฟอร์ม เชื่อมโยงภาพหรือวิดีโอด้วย โครงสร้างของโค้ดเอชทีเอ็มแอลจะอยู่ในลักษณะภายในวงเล็บสามเหลี่ยม [3]

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

HTML เป็นภาษาหลักที่ใช้ในการเขียนเว็บเพจ โดยใช้ Tag ในการกำหนดการแสดงผล HTML ย่อมาจากคำว่า Hypertext Markup Language โดย Hypertext หมายถึง ข้อความที่เชื่อมต่อกันผ่านลิงก์ (Hyperlink) Markup language หมายถึงภาษาที่ใช้ Tag ในการกำหนดการแสดงผลสิ่งต่างๆ ที่แสดงอยู่บนเว็บเพจ ดังนั้น HTML จึงหมายถึง ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่ต่างก็เชื่อมถึงกันใน Hyperspace ผ่าน Hyperlink นั่นเอง [7]

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 ซึ่ง คือ Extended HTMLซึ่งมีความสามารถและมาตรฐานที่รัดกุมกว่าอีกด้วย โดยอยู่ภายใต้การควบคุมของ W3C (World Wide Web Consortium) [8]



ภาพที่ 2.2 ไทม์ไลน์ของ HTML

### 2.3.1 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 แรกที่ถูกนำไปใช้เป็นมาตรฐานต่อไปใน ภายหน้า
- 24 พฤศจิกายน 1995: HTML 2.0 เผยแพร่เป็น RFC 1866 และบรรจุ RFC 2070 (สากล) 14 มกราคม 1997
- 12 กันยายน 1996: HTML 3.2 เผยแพร่เป็นคำแนะนำของ W3C เป็นรุ่นแรกที่ได้รับการพัฒนาและได้รับมาตรฐานโดย 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 รูปแบบ เช่นเดียวกับ 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



## 2.3.2 โครงสร้าง HTML

### 2.3.2.1 องค์ประกอบของภาษา HTML

เอกสาร html จะมีองค์ประกอบหลักๆ อยู่ 2 ส่วน คือ ส่วนที่ต้องการให้แสดงบนจอภาพและส่วนข้อความที่เป็นคำสั่ง โดยคำสั่งในเอกสาร html นี้จะเรียกว่า แท็ก (Tag)

โดยแท็กจะต้องขึ้นต้นด้วย < ตามด้วย ชื่อแท็ก ปิดท้ายด้วย > ดังนี้ <Tag name> ซึ่งจะเรียกว่าแท็กเปิดแล้วจะต้องปิดท้ายข้อความด้วยแท็กปิด ซึ่งจะมีลักษณะดังนี้ </Tag name> ชื่อแท็กต่างๆสามารถพิมพ์ตัวใหญ่หรือตัวเล็กก็ได้ความหมายเหมือนกัน

```
<Tag name> ข้อความที่ต้องการให้แสดง</Tag name>
```

### 2.3.2.2 โครงสร้างของภาษา HTML

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

```
<html>
  <head>
    <title>ข้อความที่ต้องการให้แสดงบนไต่เติ้ลบาร์ </title>
  </head>
  <body>
    คำสั่งและข้อความให้แสดงบน browser
  </body>
</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 นั้นใช้ตั้งเป็นไต่เติ้ลสำหรับบอกโดยรวมว่าเอกสารนั้นต้องการเสนออะไร แล้วเวลาที่จะ bookmark ชื่อที่จะ save คือชื่อที่อยู่ใน บล็อก title นี้ ชื่อไต่เติ้ลนี้จะต้องมีความยาวไม่เกิน 64 ตัวอักษร

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

comment (หมายเหตุ) หลังจากที่ทำกรเขียนโปรแกรมไปสักระยะ อาจจะนำโปรแกรมที่เขียนขึ้นมานั้นไปพัฒนาต่อ ซึ่งอาจจะลืมเนื้อหาสาระสำคัญแล้วทำให้ดำเนินการเขียนต่อไม่ได้ จึงต้องมีการเขียน comment เพื่อช่วยเตือนความทรงจำได้

สำหรับคำสั่ง comment จะใช้ <! เป็นแท็กเปิด และใช้ > เป็นแท็กปิดข้อความที่อยู่ระหว่าง <!--> จะไม่ถูกแสดงบน เว็บเบราว์เซอร์ [9]

## 2.4 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 ประโยชน์ของ CSS

2.4.1.1 CSS มีคุณสมบัติมากกว่า tag ของ html เช่น การกำหนดกรอบให้ข้อความ รวมทั้งสีรูปแบบของข้อความที่กล่าวมาแล้ว

2.4.1.2 CSS นั้นกำหนดที่ต้นของไฟล์ html หรือตำแหน่งอื่น ๆ ก็ได้ และสามารถมีผล กับเอกสารทั้งหมด หมายถึงกำหนด ครั้งเดียวจุดเดียวก็มีผลกับการแสดงผลทั้งหมด ทำให้เวลาแก้ไขหรือปรับปรุงทำได้สะดวก ไม่ต้องไล่ตามแก้ tag ต่างๆ ทั่วทั้งเอกสาร

2.4.1.3 CSS สามารถกำหนดแยกไว้ต่างหากจาก ไฟล์เอกสาร html และสามารถนำมาใช้ร่วมกับเอกสารหลายไฟล์ได้ การแก้ไขก็แก้เพียง จุดเดียวก็มีผลกับเอกสารทั้งหมด

CSS กับ HTML / XHTML นั้นทำหน้าที่คนละอย่างกัน โดย HTML / XHTML จะทำหน้าที่ในการวางโครงร่างเอกสารอย่างเป็นรูปแบบ ถูกต้อง เข้าใจง่าย ไม่เกี่ยวข้องกับการแสดงผล ส่วน CSS จะทำหน้าที่ในการตกแต่งเอกสารให้สวยงาม เรียกได้ว่า HTML / XHTML คือส่วน coding ส่วน CSS คือส่วน design [4]

## 2.4.2 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>
    <h1><font color="red" face="Arial">วิธีกินผลไม้ที่ถูกต้อง</font></h1>
    <p>
      <font color="black" face="Arial">
        <b>
          ให้กินผลไม้แค่ทีละอย่าง เช่นจะกินมะม่วงก็มะม่วงอย่างเดียวทั้งมือ
        </b>
      </font>
    </p>
  </body>
</html>
```

ตัวอย่างกรณีที่จัดรูปแบบการแสดงผลด้วยภาษา 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>
    <h1>วิธีกินผลไม้ที่ถูกต้อง</h1>
    <p>ให้กินผลไม้แค่ทีละอย่าง เช่นจะกินมะม่วงก็มะม่วงอย่างเดียวทั้งมือ</p>
  </body>
</html>

```

ด้านบนคือตัวอย่างเมื่อเปลี่ยนมาใช้คำสั่ง style sheet จัดรูปแบบการแสดงผลแทนการใช้ code ภาษา HTML ทำให้ code ภายในเอกสารอ่านเข้าใจง่าย และแก้ไขได้ง่ายขึ้น [10]

## 2.5 JavaScript

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง Java JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์มีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียนเต็ลเต็ด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server) [5]

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

เพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเนตสเคปจึงได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ ปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อกับภาษาจาวาได้ 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 [12]

## 2.5.1 ความสามารถของ JavaScript

2.5.1.1 JavaScript ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายๆได้ โดยไม่ต้องพึ่งภาษาอื่น

2.5.1.2 JavaScript มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่นเมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox ก็สามารถสั่งให้เปิดหน้าต่างใหม่ได้ ทำให้เว็บไซต์ของเรามีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น นี่คือข้อดีของ JavaScript เลยก็ว่าได้ที่ทำให้เว็บไซต์ต่างๆทั้งหลายเช่น Google Map ต่างหันมาใช้

2.5.1.3 JavaScript สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้ นั่นคือสามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้ หรือหน้าแสดงเนื้อหาสามารถซ่อนหรือแสดงเนื้อหาได้แบบง่ายๆนั่นเอง

2.5.1.4 JavaScript สามารถใช้ตรวจสอบข้อมูลได้ สังเกตว่าเมื่อเรากรอกข้อมูลบางเว็บไซต์ เช่น Email เมื่อเรากรอกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่าเรากรอกผิด หรือลิ้มกรอกอะไรบางอย่าง เป็นต้น

2.5.1.5 JavaScript สามารถใช้ในการตรวจสอบผู้ใช้ได้เช่น ตรวจสอบว่าผู้ใช้ ใช้ web browser อะไร

2.5.1.6 JavaScript สร้าง Cookies (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

## 2.5.2 ข้อดีและข้อเสียของ Java JavaScript

การทำงานของ JavaScript เกิดขึ้นบนเบราว์เซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้เบราว์เซอร์อะไร หรือที่ไหน ก็ยังสามารถใช้ 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) [11]

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 2.3 การใช้งาน JavaScript ในแท็ก <body>

## 2.5.3 รู้จักกับตัวแปรใน Javascript

**Global Variable** เป็นตัวแปรที่สามารถเรียกใช้ได้ทุกที่ นั่นคือประกาศไว้นอกฟังก์ชัน **Local Variable** เป็นตัวแปรที่สามารถเรียกได้เฉพาะในฟังก์ชันใดฟังก์ชันหนึ่ง ที่มีการกำหนดตัวแปรไว้เท่านั้น นั่นคือได้ประกาศตัวแปรไว้ภายในฟังก์ชัน

## 2.5.4 การประกาศตัวแปร var let และ const ใช้งานต่างกันอย่างไร

2.5.4.1 var เป็นการประกาศตัวแปรแบบ global scope คือ ประกาศค่าและใช้ค่าร่วมกันได้

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

2.5.4.2 let เป็นการประกาศตัวแปรแบบ block scope

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

2.5.4.3 Const เป็นการประกาศตัวแปรแบบ block scope ไม่สามารถเปลี่ยนแปลงค่าได้แต่การเปลี่ยนค่าภายใน 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 }
```

### 2.5.5 Array ชุดของข้อมูล

เป็นตัวแปรพิเศษซึ่งสามารถเก็บมากกว่าหนึ่งค่าในแต่ละครั้งหากต้องการมีรายการสินค้า (ตัวอย่างเช่นชื่อรถ) การจัดเก็บรถยนต์ในตัวแปรเดี่ยวอาจมีลักษณะดังนี้:

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

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 accesses 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 2.4 ตัวอย่างชุดอาเรย์ Cars

#### JavaScript Arrays

JavaScript array elements are accesses using numeric indexes (starting from 0).

Saab

ภาพที่ 2.5 Index Car[0] = Saab



### 2.5.6 function () ฟังก์ชัน

ฟังก์ชัน หรือชุดคำสั่งที่รวม Statement การทำงานเอาไว้ สามารถเรียกชื่อมาใช้งานตามที่ต้องการใช้ได้

```
function ชื่อฟังก์ชัน( ตัวแปรพารามิเตอร์ถ้ามี ){Statement ที่ต้องการให้ทำงาน เมื่อมีการเรียก
functionreturn <- ถ้าต้องการส่งค่าบางอย่างกลับไปนอกฟังก์ชัน เมื่อการทำงานมาถึงจุดนี้
}
```

ตัวอย่างการเรียกใช้ฟังก์ชัน เมื่อสร้างไว้เบื้องต้น

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

### 2.5.7 Loop ใช้ทำซ้ำตามเงื่อนไขที่กำหนด Loop for

```
for ( ตัวแปรเริ่มต้น ; กรณีเทียบ (Condition) ; เพิ่มค่าตัวแปร ) {
    Statement ที่ต้องการให้ทำงาน ถ้ากรณีเทียบยังเป็นจริงอยู่
}
```

ตัวอย่างการทำงาน Loop for

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

## 2.6 XAMPP

Xampp เป็นโปรแกรม Apache web server ไว้จำลอง web server เพื่อไว้ทดสอบ สคริปหรือเว็บไซต์ในเครื่อง โดยที่ไม่ต้องเชื่อมต่ออินเทอร์เน็ตและไม่ต้องมีค่าใช้จ่ายใดๆ ง่ายต่อการติดตั้งและใช้งานโปรแกรม Xampp จะมาพร้อมกับ PHP ภาษาสำหรับพัฒนาเว็บแอปพลิเคชันที่เป็นที่นิยม, MySQL ฐานข้อมูล, Apache จะทำหน้าที่เป็นเว็บ เซิร์ฟเวอร์, Perl อีกทั้งยังมาพร้อมกับ OpenSSL , phpMyadmin (ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล สนับสนุนฐานข้อมูล MySQL และ SQLite โปรแกรม Xampp จะอยู่ในรูปแบบของไฟล์ Zip, tar, 7z หรือ exe โปรแกรม Xampp อยู่ภายใต้ใบอนุญาตของ GNU General Public License แต่บางครั้งอาจจะมีการเปลี่ยนแปลงเรื่องของลิขสิทธิ์ในการใช้งาน จึงควรติดตามและตรวจสอบโปรแกรมด้วย

### 2.6.1 โปรแกรม XAMPP สามารถใช้งานได้ 4 OS ได้แก่

- Windows สามารถใช้งานได้กับ 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

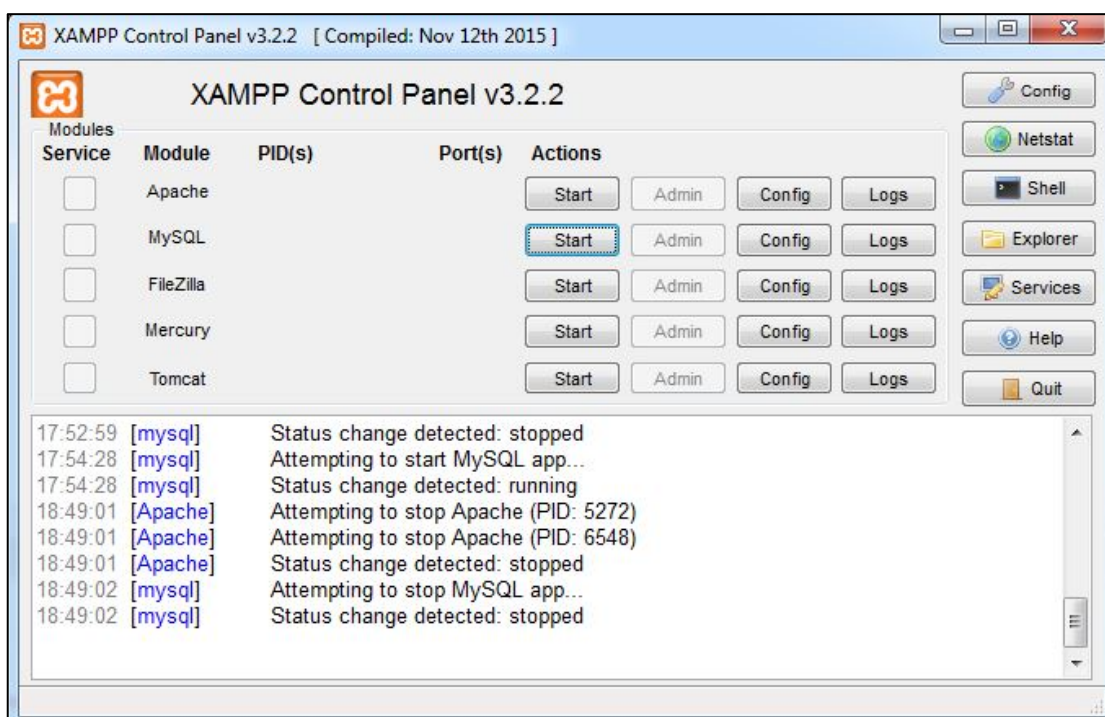
### 2.6.2 ข้อกำหนดด้านเทคนิค

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

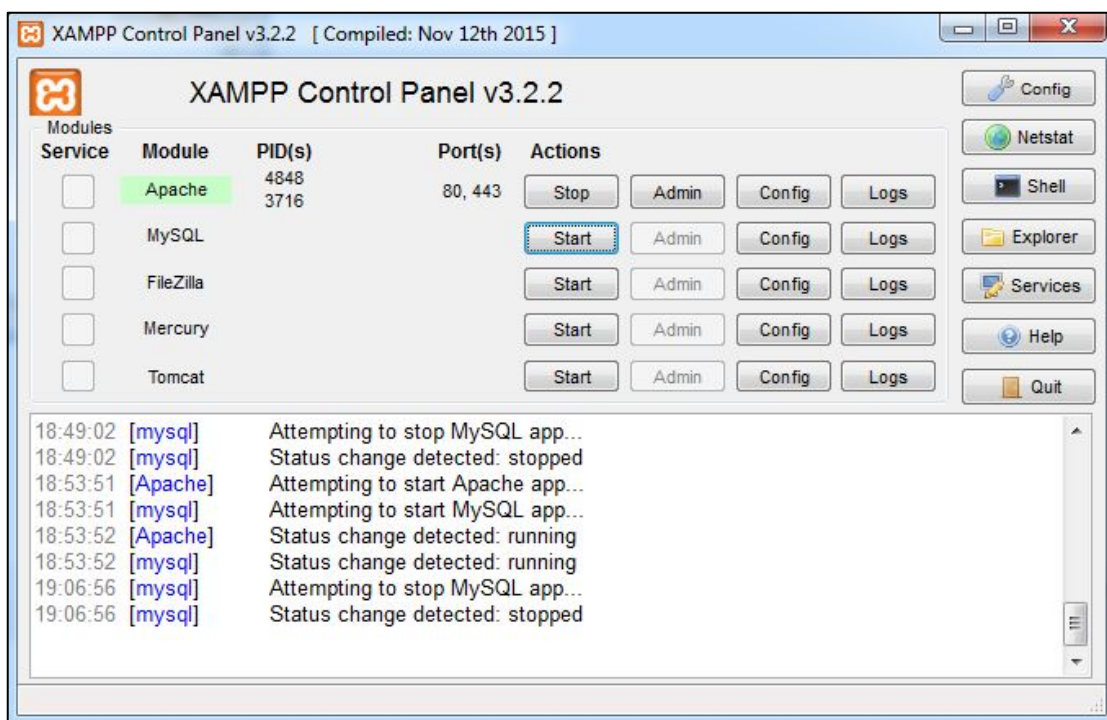
จะเห็นได้ว่า Xampp เป็นโปรแกรม Apache web server ที่ได้รับความนิยมมากในปัจจุบันและยังมีการพัฒนาอย่างต่อเนื่อง โดยเวอร์ชันล่าสุดของ Xampp คือ PHP 7.3.9 [13]

### 2.6.3 ขั้นตอนการเปิดใช้งานเซิร์ฟเวอร์

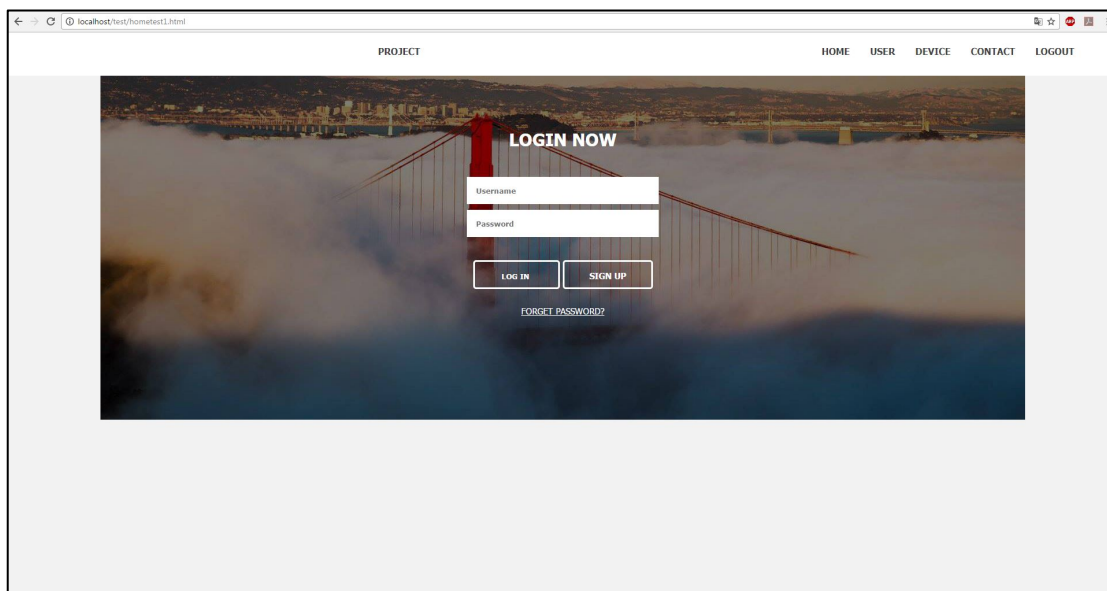
การเปิดใช้งาน Xampp เพื่อจำลองเซิร์ฟเวอร์ สำหรับการจำลองเว็บไซต์เสมือนการใช้งานจริงบนเซิร์ฟเวอร์ ซึ่งสามารถดาวน์โหลด Xampp ตัวติดตั้งได้ฟรีที่ เว็บไซต์ [apachefriends.org](https://www.apachefriends.org) หรือเข้าไปที่ URL : <https://www.apachefriends.org/index.html> เมื่อติดตั้งแล้วสามารถทำตามภาพประกอบด้านล่างเพื่อเปิดใช้งาน



ภาพที่ 2.6 เปิดโปรแกรม Xampp

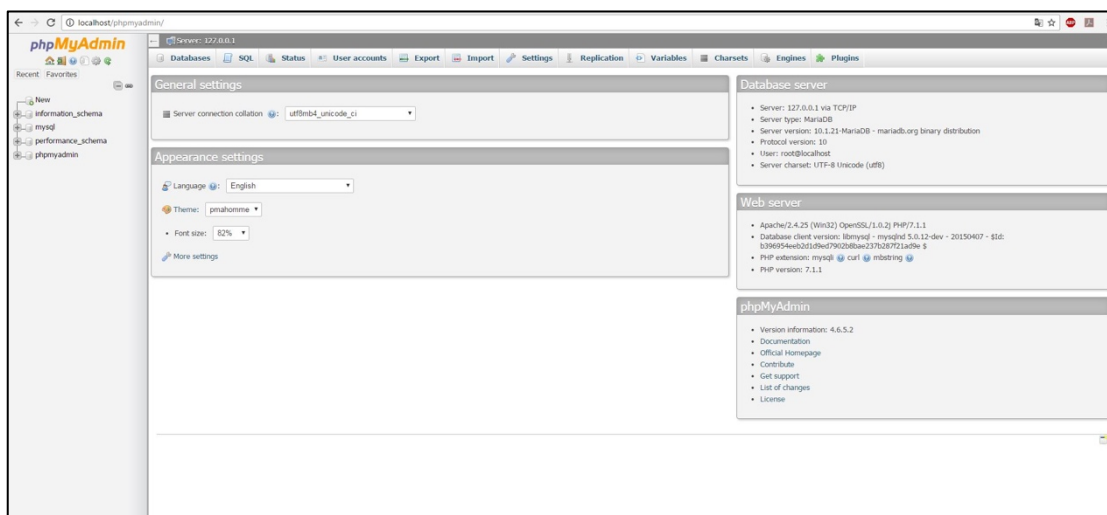


ภาพที่ 2.7 กดปุ่ม start บน module Apache



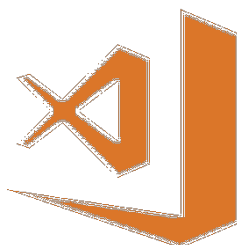
ภาพที่ 2.8 เปิด website ที่ต้องการจำลอง server

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



ภาพที่ 2.9 phpMyAdmin สร้างฐานข้อมูล

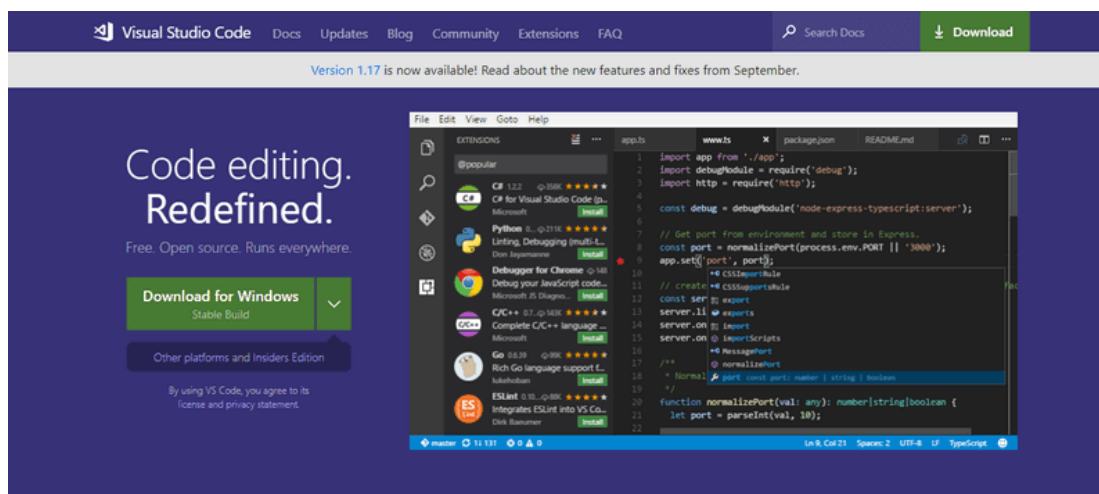
## 2.7 Visual Studio Code



ภาพที่ 2.10 Visual Studio Code

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

ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์มรองรับการใช้งานทั้งบน 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 2.11 Code editing

ความแตกต่างระหว่าง VSCode และ Visual Studio คือ

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

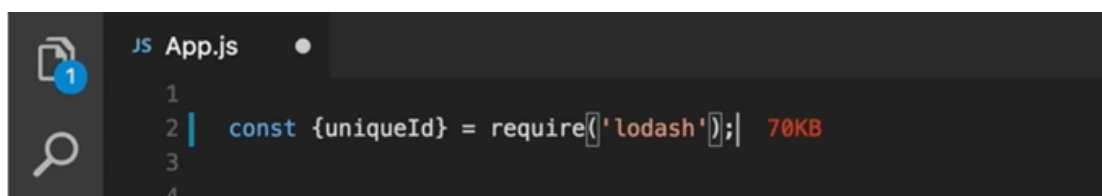
สำหรับผู้สนใจใช้งาน สามารถดาวน์โหลดได้ที่ <https://code.visualstudio.com/>

### 2.7.1 Visual Studio Code Extensions

ตัว Visual Studio Code ยังมีตัวช่วยเสริม เพื่อให้นักพัฒนาสามารถใช้งานได้ง่ายขึ้น โดยจะยกตัวอย่าง Extensions ให้พอเห็นภาพโดยรวมดังนี้

#### 2.7.1 Import Cost

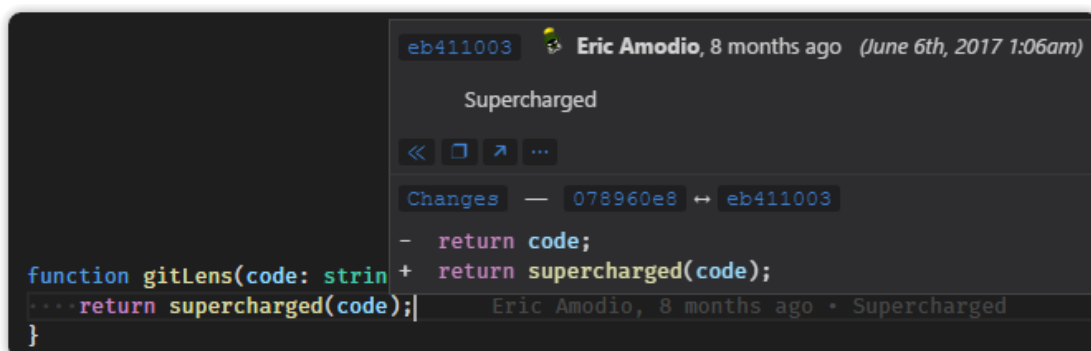
Import cost เป็น extension ที่จะแสดงขนาดไฟล์ของ package ที่ import เข้ามา โดยจะแสดงให้เห็นทันทีว่า package ตัวไหนมีขนาดไฟล์ที่มากเกินไป



ภาพที่ 2.12 Import Cost

### 2.7.2 GitLens — Git supercharged

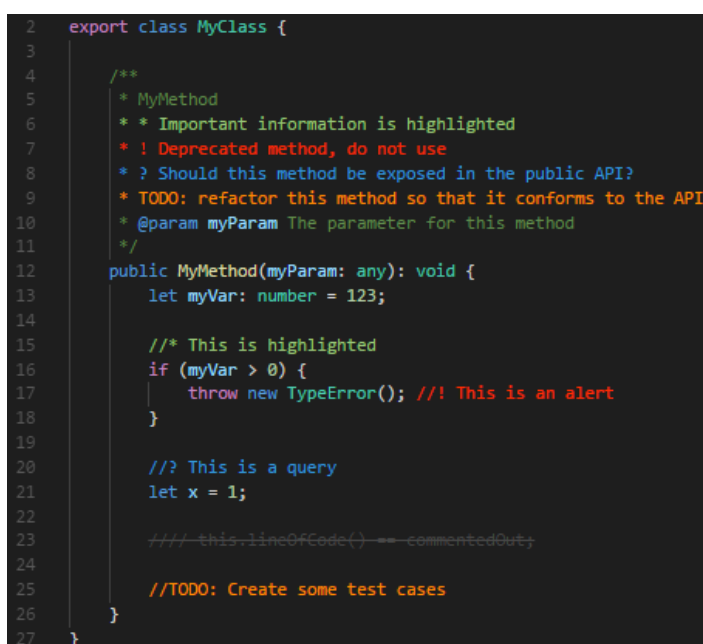
ถึงแม้ว่า Visual Studio Code จะมีการฝัง Git เข้ามาในตัว editor อยู่แล้ว แต่ก็สามารถที่จะเพิ่มพลังให้มันได้อีกด้วย GitLens โดยเมื่อเอา cursor ไปวางไว้ที่โค้ด ก็จะได้เห็นว่าใครเป็นคน commit โค้ดส่วนนั้นขึ้นไป เมื่อไหร่ แล้วยังดู changes ย้อนหลังได้อีกด้วย



ภาพที่ 2.13 GitLens — Git supercharged

### 2.7.3 Better Comments

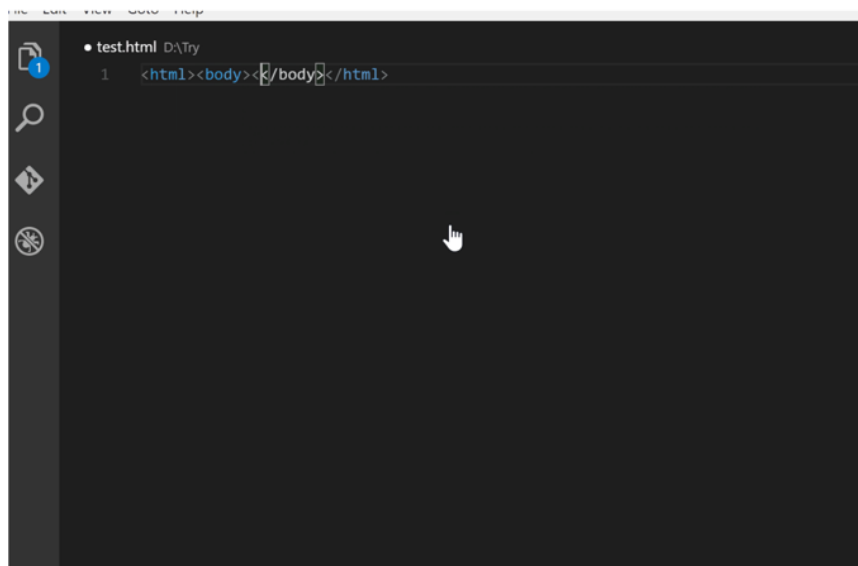
บางทีการเขียน TODO ก็อาจจะกลืนไปกับ comment ทั่วไปมากเกินไปจนแยกไม่ออกว่าอันไหนคือ comment อันไหนคือ TODO แต่ Better Comments จะช่วยแยกสีให้ชัดเจน



ภาพที่ 2.13 Better Comments

### 2.7.4 Auto Close Tag

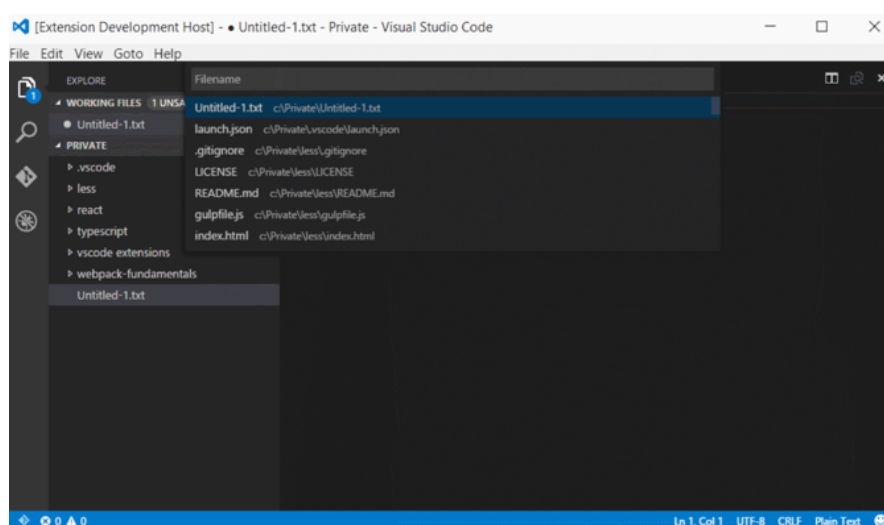
เวลาเขียนโค้ดบางทีก็อยากทำงานเร็วๆ เขียนแค่ tag เปิดก็พอ ซึ่ก็เขียน tag ปิด ควรที่จะลง Auto Close Tag ไว้ให้ editor ทำการปิด tag ให้อัตโนมัติ



ภาพที่ 2.14 Auto Close Tag

### 2.7.5 Relative Path

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

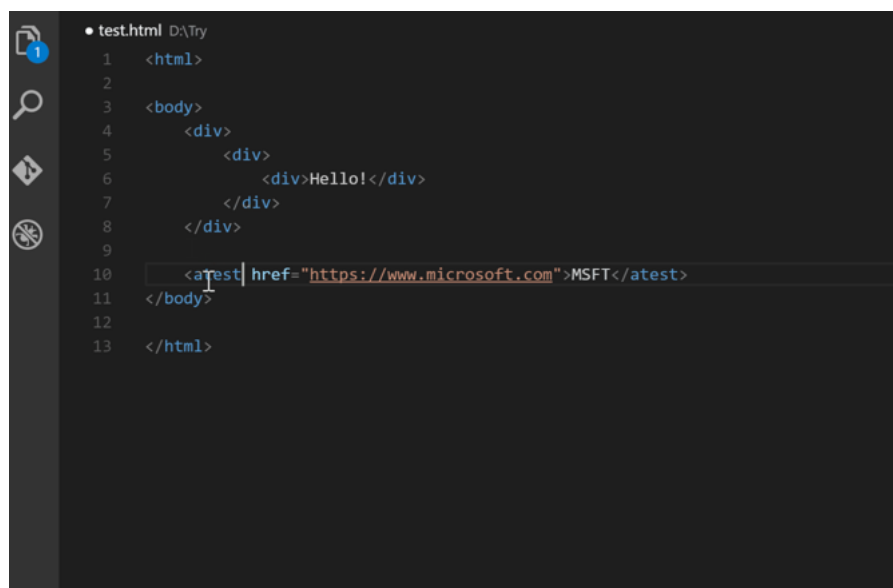


ภาพที่ 2.15 Auto Close Tag



### 2.7.6 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 2.16 Auto Close Tag

### 2.7.7 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 2.17 npm intellisense

### 2.7.8 Bracket Pair Colorizer

หลายๆ คนมักจะเจอปัญหาว่า ลืมใส่วงเล็บ หรือมีปีกกาเกินมา อยู่เป็นประจำ แลบบางทีก็ดูไม่ค่อยออกว่าอะไรจับคู่กับอะไรอยู่ 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 2.18 Bracket Pair Colorizer

### 2.7.9. Quokka.js

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

```

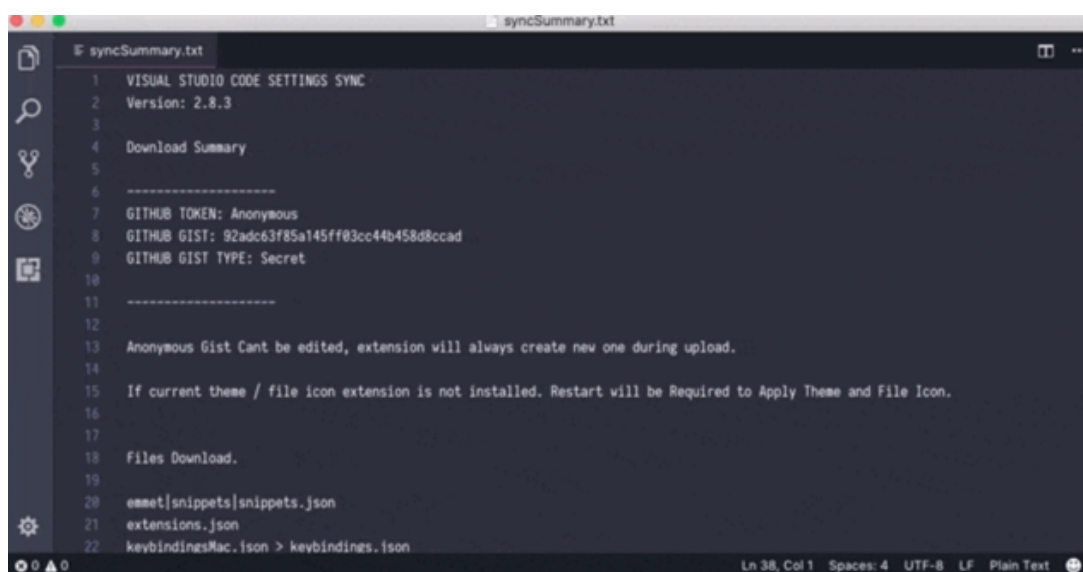
Untitled-1
1  import resolveAfter50ms from './resolve'
2
3  async function add(x) {
4      if (x < 0) throw new Error('The number must be positive');
5      var a = resolveAfter50ms(60);
6      var b = resolveAfter50ms(30);
7      return x + await a + await b;
8  }
9
10 add(10).then(result => {
11  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

```

ภาพที่ 2.19 Quokka.js

### 2.7.10 Settings Sync

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



ภาพที่ 2.19 Settings Sync

## 2.8 การประเมินความพึงพอใจ

### 2.7.1 การสร้างแบบประเมินความพึงพอใจของระบบ

หลังจากการพัฒนาเว็บจอสอบโพรเจกออนไลน์ ผู้จัดทำได้ทำการเพิ่มข้อมูลอาจารย์และนักศึกษา รวมถึงข้อมูลอื่นที่เกี่ยวข้องเพื่อทดสอบระบบด้วยแบบประเมินความพึงพอใจ

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

ตารางที่ 2-2 เกณฑ์การให้คะแนนของแบบประเมินความพึงพอใจ

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

ตารางที่ 2-3 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