บทที่ 10
การเชื่อมโยงระหว่างเว็บเพจ คุกกี้ และเซสชัน

การเชื่อมต่อระหว่างบราวเซอร์และเว็บเซิร์ฟเวอร์นั้น จะเกิดขึ้นเมื่อบราวเซอร์ส่งการร้องขอ (Request) ออกไป และเมื่อเว็บเซิร์ฟเวอร์ส่งข้อมูลผลลัพธ์กลับมาครบทั้งหมดแล้ว การเชื่อมต่อก็จะสิ้นสุดลง จะส่งผลให้ข้อมูลต่างๆ ที่ถูกสร้างขึ้นภายในเว็บเพจที่เรียกใช้ในขณะนั้นถูกทำลายลงไปด้วย แต่ถ้าข้อมูลที่ถูกสร้างขึ้นในเว็บเพจนี้ และมีความจำเป็นต้องนำไปใช้ในเว็บเพจอื่นๆ ด้วย ทำอย่างไรจึงจะเก็บรักษาข้อมูลนั้นเอาไว้ได้ ดังนั้นในบทนี้จะมาศึกษาถึงวิธีการที่จะจัดเก็บข้อมูลเพื่อให้สามารถนำไปใช้งานระหว่างเว็บเพจ ที่จะกล่าวถึงในบทนี้ คือ การเชื่อมโยงระหว่างเว็บเพจ คุกกี้และเซสชัน

10.1 เฮดเดอร์

เฮดเดอร์ (Header) คือ ข้อมูลบางอย่างที่ใช้ในการสื่อสารกันระหว่างเว็บเซิร์ฟเวอร์ และบราวเซอร์ เฮดเดอร์อาจจะเป็นข้อมูลที่ส่งมาจากเว็บเซิร์ฟเวอร์ไปยังบราวเซอร์ หรือส่งจากบราวเซอร์ไปยังเว็บเซิร์ฟเวอร์ก็ได้ โดยวัตถุประสงค์ของการส่งเฮดเดอร์นั้นมีหลายลักษณะ การกำหนดเฮดเดอร์จะใช้ฟังก์ชัน header ( ) มีรูปแบบดังนี้

รูปแบบ

```
void header ( string $string [, bool $replace = true [, int $http_response_code ]] )
```

เมื่อ $string หมายถึง ชื่อเฮดเดอร์ มีหลายลักษณะขึ้นอยู่กับว่าจะส่งข้อมูลอะไรไปยังบราวเซอร์

$replace หมายถึง การแทนที่ ค่าโดยปริยายคือ true

$http_response_code หมายถึง ลักษณะการตอบสนอง จะต้องกำหนดให้อยู่ในสตริงเดียวกันกับชื่อเฮดเดอร์

ตัวอย่างที่ 10.1 การย้ายหน้าเว็บเพจ (Webpage Redirect) ไปยัง URL อื่น ๆ

```
1   <?php
2       header ('Location: http://www.freebsd.sru.ac.th/');
3   ?>
```

จากตัวอย่างที่ 10.1 การย้ายหน้าเว็บเพจ (Webpage Redirect) ไปยัง URL อื่น ๆ โดยใช้ฟังก์ชัน header ( ) โดยพารามิเตอร์ที่ระบุภายในฟังก์ชัน คือ URL ที่ต้องการย้ายหรือไปทำงานต่อ

ตัวอย่างที่ 10.2 การย้ายหน้าเว็บเพจแบบ refresh (หน่วงเวลาก่อนเปลี่ยนหน้าเว็บเพจ มีหน่วยเป็นวินาที)

```
1    <?php
2        header ('Refresh: 10; url=http:// www.freebsd.sru.ac.th/');
3    ?>
```

จากตัวอย่างที่ 10.2 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน header ( ) สำหรับย้ายหน้าเว็บเพจแบบกำหนดพารามิเตอร์ refresh (หน่วงเวลาก่อนเปลี่ยนหน้าเว็บเพจ มีหน่วยเป็นวินาที) และกำหนดเวลา 10 นาที ก่อนทำการย้าย

ตัวอย่างที่ 10.3 การกำหนดภาษาที่ใช้ในหน้าเว็บ

```
1    <?php
2        header ('Content-language: en');
3    ?>
```

จากตัวอย่างที่ 10.3 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน header ( ) สำหรับกำหนดภาษาที่ใช้แสดงภายในหน้าเว็บเพจ จากตัวอย่างเป็นการกำหนดให้แสดงผลเป็นภาษาอังกฤษ

ตัวอย่างที่ 10.4 การใช้ฟังก์ชัน header ( ) สำหรับสร้างไฟล์สำหรับการดาวน์โหลด

```
1    <?php
2        header ('Content-Type: application/octet-stream');
3        header ('Content-Disposition: attachment; filename="example.zip"');
4        header ('Content-Transfer-Encoding: binary');
5    ?>
```

จากตัวอย่างที่ 10.4 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน header ( ) สำหรับสร้างไฟล์สำหรับการดาวน์โหลด

ตัวอย่างที่ 10.5 การสร้างเว็บเพจโดยไม่ให้มีการเขียนเว็บแคช (web cache)

```
1    <?php
2        header ('Cache-Control: no-cache, no-store, max-age=0, must-revalidate');
3        header ('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
4    ?>
```

จากตัวอย่างที่ 10.5 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน header ( ) สำหรับการสร้างเว็บเพจโดยไม่ให้มีการเขียนเว็บแคช เว็บแคชทำให้เข้าถึงเว็บไซต์ได้เร็วยิ่งขึ้น เพราะจะไม่มีการประมวลผลใดๆ ของหน้าเว็บไซต์ที่ผู้ใช้งานเรียก แต่อย่างไรก็ตามเว็บแคช ก็ยังมีข้อเสียในการทำงาน นั่นคือข้อมูลที่ได้ อาจจะไม่อัพเดต เพราะฉะนั้นระบบเว็บแคชจึงจำเป็นที่จะต้องมีการตั้งเวลาให้หมดอายุของแคช และทำการอัพเดตแคชใหม่ ถ้าหากว่าเวลาของแคชเยอะเกินไป ก็จะทำให้ข้อมูลอัพเดตช้าไปด้วย

ตัวอย่างที่ 10.6 การกำหนดประเภทของการแสดงเนื้อหาของเว็บ (web content)

```
1   <?php
2       header ('Content-Type: text/html; charset=iso-8859-1');
3       header ('Content-Type: text/html; charset=utf-8');
4       header ('Content-Type: text/plain');          // plain text file
5       header ('Content-Type: image/jpeg');          // JPG picture
6       header ('Content-Type: application/zip');     // ZIP file
7       header ('Content-Type: application/pdf');     // PDF file
8       header ('Content-Type: audio/mpeg');          // Audio MPEG (MP3,...) file
9   ?>
```

จากตัวอย่างที่ 10.6 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน header ( ) สำหรับการกำหนดประเภทของการแสดงเนื้อหาของเว็บ

10.2 การส่งข้อมูลระหว่างเว็บเพจแบบ Query String

Query String คือ รูปแบบการส่งข้อมูลจากเว็บเพจหนึ่งไปยังอีกเว็บเพจหนึ่ง โดยแนบข้อมูลต่อท้าย URL มีตัวอย่างดังต่อไปนี้

```
http://localhost/sru_rpm/member.php?username=admin&board_id=123&group=99
```

ภาพที่ 10.1 แสดงตัวอย่างรูปแบบของ Query String

จากตัวอย่างที่ 10.1 แสดงตัวอย่างรูปแบบ Query String จะเห็นได้ว่าตัวแปรและค่าของตัวแปรแต่ละชุดจะใช้สัญลักษณ์คั่นเป็นช่วงๆ ตามลำดับ โดยมีรายละเอียดของสัญลักษณ์ ดังตารางที่ 10.1

ตารางที่ 10.1 สัญลักษณ์ที่ใช้ร่วมกับ Query String เพื่อการส่งค่าระหว่างเว็บเพจ

สัญลักษณ์	ความหมาย
?	ใช้คั่น URL กับข้อมูลที่จะแนบไป
&	ใช้คั่นตัวแปรแต่ละตัว
=	ใช้คั่นระหว่างตัวแปรและค่าของตัวแปร เช่น password=123
+	ใช้แทนช่องว่าง 1 ช่องของชุด URL เช่น keyword=parinya+noidonprai
%	ใช้แทนอักขระพิเศษที่ไม่ใช่ตัวอักษรหรือตัวเลข โดยจะใช้เครื่องหมาย % นำหน้าเลขฐาน 16 ที่จะใช้แทนอักขระตัวนั้นๆ เช่น %28 ใช้แทนเครื่องหมายวงเล็บเปิด เป็นต้น

สำหรับการอ่านข้อมูลที่เว็บเพจปลายทางที่ได้รับข้อมูล Query String สามารถเรียกใช้ได้หลายลักษณะ ดังนี้

10.2.1 ใช้ตัวแปร $_GET เหมือนกับการอ่านข้อมูลที่ส่งจากฟอร์มด้วยวิธี GET ตามปกติ เช่น จากภาพที่ 10.1 $member = $_GET ["username"]; หมายความว่า กำหนดให้ตัวแปร $member มีค่าเท่ากับ admin เพราะ $_GET ["username"] มีค่าเท่ากับ admin

10.2.2 ถ้าต้องการอ่าน Query String ทั้งหมดออกมาจาก URL เพื่อส่งต่อไปยังเว็บเพจอื่นๆ ก็สามารถอ่านจากตัวแปร $_SERVER ['Query_STRING'] ดังตัวอย่างที่ 10.7

ตัวอย่างที่ 10.7 การอ่าน Query String ด้วย $_SERVER ['Query_STRING']

```
1    <?php
2        $qrystr = $_SERVER ['Query_STRING'];
3        $goto = "nextpage.php" . "?" . $qrystr;
4        header ("Location: $goto");
5    ?>
```

จากตัวอย่างที่ 10.7 การอ่าน Query String ทั้งหมดออกมาจาก URL ด้วย $_SERVER ['Query_STRING'] อธิบายร่วมกับภาพที่ 10.1 ดังนี้

บรรทัดที่ 2 จะหมายความว่า กำหนดให้ตัวแปร $qrystr มีค่าเท่ากับ "username= admin &board_id=123&group=99"

บรรทัดที่ 3 กำหนดให้ตัวแปร $goto มีค่าเท่ากับ "nextpage.php?username= admin &board_id=123&group=99" (คือ การต่อข้อความให้เป็นประโยคเดียวกัน)

บรรทัดที่ 4 ใช้ฟังก์ชัน header ( ) เพื่อย้ายหน้าเว็บเพจปัจจุบันไปยังเว็บเพจ nextpage.php พร้อมด้วยค่า Query String เพื่อนำค่าทั้งหมดไปประมวลผลต่อ

10.2.3 ใช้ฟังก์ชัน parse_str ( ) จะได้ผลลัพธ์เป็นตัวแปรพารามิเตอร์ และค่าของตัวแปรนั้นๆ มีตัวอย่างการใช้งาน ดังนี้

ตัวอย่างที่ 10.8 การอ่าน Query String ทั้งหมดออกมาจาก URL ด้วยฟังก์ชัน parse_str ( )

```
1   <?php
2       $qrystr = $_SERVER ['Query_STRING'];
3       parse_str ($qrystr);
4       echo "$username <br>";
5       echo "$board_id <br>";
6       echo "$group <br>";
7   ?>
```

ผลลัพธ์

```
admin
123
99
```

จากตัวอย่างที่ 10.8 การอ่าน Query String ทั้งหมดออกมาจาก URL ด้วยฟังก์ชัน parse_str ( ) อธิบายร่วมกับภาพที่ 10.1 ดังนี้

บรรทัดที่ 2 จะหมายความว่า กำหนดให้ตัวแปร $qrystr มีค่าเท่ากับ "username= admin &board_id=123&group=99"

บรรทัดที่ 3 ใช้ฟังก์ชัน parse_str ($qrystr) ผลของฟังก์ชันจะได้ตัวแปรพร้อมใช้งานในหน้าเว็บเพจที่ใช้ฟังก์ชัน ประกอบด้วยตัวแปร ดังนี้

บรรทัดที่ 4 แสดงผลค่าตัวแปร $username

บรรทัดที่ 5 แสดงผลค่าตัวแปร $board_id

บรรทัดที่ 6 แสดงผลค่าตัวแปร $group

10.3 การจัดเก็บข้อมูลแบบคุกกี้

คุกกี้ (Cookie) เป็นรูปแบบของการเก็บรักษาข้อมูลบางอย่างไว้บนเครื่องของผู้ใช้ เพื่อจะนำข้อมูลนี้กลับมาใช้ใหม่ภายหลังได้ โดยข้อมูลที่เก็บในแบบคุกกี้ไม่ควรเป็นข้อมูลที่ต้องเก็บเป็นความลับ เช่น รหัสผ่าน เป็นต้น เนื่องจากข้อมูลคุกกี้จะเก็บไว้ในรูปแบบของไฟล์ข้อความธรรมดา ระดับความปลอดภัยจึงค่อนข้างต่ำ แต่อย่างไรก็ตาม เนื่องจากผู้ใช้บางกลุ่มกลับเห็นว่า คุกกี้นั้นเป็นการล่วงละเมิดสิทธิของผู้ใช้ ดังนั้นโปรแกรมเว็บบราวเซอร์จึงมีตัวเลือกสำหรับให้ผู้ใช้ สามารถปฏิเสธการบันทึกคุกกี้จากเว็บไซต์ที่ไม่ต้องการได้ ด้วยเหตุนี้การใช้คุกกี้จึงอาจไม่บรรลุผลเสมอไป

นอกจากการปฏิเสธคุกกี้จากผู้ใช้แล้ว ยังมีสาเหตุอื่นๆ ที่อาจทำให้ไม่สามารถใช้คุกกี้ได้ เช่น การหมดอายุของคุกกี้ หรือไฟล์ที่จัดเก็บคุกกี้อาจเสียหาย ใช้การไม่ได้ หรือถูกทำลาย เช่น การติดตั้งระบบใหม่ เป็นต้น ดังนั้นต้องจัดเตรียมทางเลือกไว้เผื่อกรณีนี้ด้วย

10.3.1 การสร้างคุกกี้

ฟังก์ชันหลักในการใช้งานเกี่ยวกับคุกกี้ คือ ฟังก์ชัน setcookie ( ) มีรูปแบบดังนี้

รูปแบบ

```
bool setcookie ( string $name [, string $value [, int $expire = 0 [, string $path [, string $domain]]]])
```

เมื่อ $name หมายถึง ชื่อคุกกี้ที่ต้องการสร้าง

$value หมายถึง ค่าที่จะกำหนดให้กับคุกกี้

$expire หมายถึง อายุของคุกกี้

$path หมายถึง พาธบนเซิร์ฟเวอร์ที่คุกกี้สามารถใช้งานได้

$domain หมายถึง กำหนดโดเมนที่สามารถอ่านคุกกี้ได้

ตัวอย่างที่ 10.9 การใช้ฟังก์ชัน setcookie ( )

```
1   <?php
2       $value = "something from somewhere";
3       setcookie ("TestCookie", $value);
4       setcookie ("TestCookie", $value, time()+3600);  /* หมดอายุใน 1 ชั่วโมง */
5       setcookie ("TestCookie", $value, time()+3600, "/~rasmus/", "example.com", 1);
6   ?>
```

จากตัวอย่างที่ 10.9 ตัวอย่างการใช้ฟังก์ชัน setcookie ( ) กรณีกำหนดชื่อคุกกี้ซ้ำกับชื่อที่มีอยู่แล้ว จะเป็นการแทนที่ข้อมูลเดิม หรือเท่ากับเป็นการแก้ไขค่าของคุกกี้นั่นเอง สำหรับรูปแบบของฟังก์ชัน setcookie ( ) นี้เป็นแบบอย่างง่ายเท่านั้น ในความเป็นจริงยังสามารถกำหนดพารามิเตอร์เพิ่มเติมได้อีกหลายรูปแบบ

10.3.2 การอ่านข้อมูลจากคุกกี้

การอ่านคุกกี้จะใช้ตัวแปร $_COOKIE เป็นตัวแปรแบบอาร์เรย์ โดยระบุชื่อของคุกกี้เป็นค่าคีย์ ก็จะได้ค่าของคุกกี้ ที่กำหนดไว้ตอนสร้างคุกกี้ แต่อย่างไรก็ตาม หากคุกกี้นั้นไม่มีอยู่จริง หรือหมดอายุไปแล้ว การอ่านคุกกี้นั้นโดยไม่ตรวจสอบก่อนจะเกิดข้อผิดพลาดขึ้น ดังนั้นเพื่อให้แน่ใจว่ามีคุกกี้นั้นอยู่จริงหรือไม่ ควรตรวจสอบด้วยฟังก์ชัน isset ( ) ก่อนอ่านค่าเสมอ ตัวอย่างดังนี้

ตัวอย่างที่ 10.10 ตัวอย่างการตรวจสอบตัวแปรคุกกี้ก่อนการอ่านค่าจากคุกกี้

```
1    <?php
2        if (isset ($_COOKIE ["book"])) {
3            $b = $_COOKIE ["book"];
4        }
5    ?>
```

จากตัวอย่างที่ 10.10 ตัวอย่างการตรวจสอบตัวแปรคุกกี้ก่อนการอ่านค่าจากคุกกี้ เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นได้

10.3.3 การกำหนดอายุ และลบคุกกี้

ปกติแล้วหากสร้างคุกกี้ด้วยวิธีการตามที่กล่าวมาแล้ว คือ ไม่ได้กำหนดอายุ จะทำให้คุกกี้หมดอายุทันทีที่ปิดบราวเซอร์ แต่โดยทั่วไปมักจะเก็บข้อมูลคุกกี้นี้ไว้ใช้ในคราวต่อๆ ไปด้วย

การกำหนดอายุของคุกกี้จะกำหนดเป็นเวลาในหน่วยวินาทีโดยเทียบกับเวลาปัจจุบัน โดยทั่วไปแล้วมักอ้างอิงเวลาปัจจุบันด้วยฟังก์ชัน time ( ) ฟังก์ชันนี้จะคืนค่า timestamp ของเวลาปัจจุบัน ดังนั้นสามารถนำอายุของคุกกี้มาบวกเพิ่มเข้าไปได้เลย เช่น time ( ) + 30 หมายความว่ากำหนดให้คุกกี้หมดอายุภายใน 30 วินาที หลังจากนี้ หรือหากต้องการกำหนดเวลาในหน่วยที่ยาวนานกว่านี้ต้องนำมาแปลงเป็นวินาที เช่น หากต้องการให้คุกกี้มีอายุ 1 วัน ต้องเขียนเป็น time ( ) + (24 * 60 * 60) เป็นต้น มีตัวอย่างดังนี้

ตัวอย่างที่ 10.11 การกำหนดอายุของคุกกี้

```
1    <?php
2        $expire = time ( ) + (24 * 60 * 60);
3        setcookie ("mycookie", "Parinya Noidonprai", $expire);
4    ?>
```

จากตัวอย่างที่ 10.11 การกำหนดอายุของคุกกี้ให้มีอายุ 1 วัน แต่หากต้องการลบคุกกี้ที่ชื่อ "mycookie" สามารถประยุกต์ตามตัวอย่างที่ 10.12 ดังนี้

ตัวอย่างที่ 10.12 การลบคุกกี้

```
1    <?php
2        $expire = time ( ) - 3600;      // ลบด้วยเลขอะไรก็ได้เพื่อให้เป็นเวลาที่ผ่านมาแล้ว
3        setcookie ("mycookie", "", $expire);
4    ?>
```

จากตัวอย่างที่ 10.12 การลบคุกกี้ โดยใช้เทคนิควิธีการย้อนเวลาเพื่อให้คุกกี้หมดเวลา หรือความหมายคือการลบคุกกี้ออกนั่นเอง

10.4 การจัดเก็บข้อมูลแบบเซสชัน

เซสชัน (Session) เป็นการจัดเก็บข้อมูลบางอย่างไว้ชั่วคราวในฝั่งของเว็บเซิร์ฟเวอร์ เพื่อให้สามารถนำข้อมูลกลับมาใช้งานใหม่หลังจากที่เว็บเพจนั้นสิ้นสุดการทำงานไปแล้ว และยังสามารถนำไปใช้งานที่เว็บเพจอื่นๆ ได้

10.4.1 ลักษณะและการทำงานของเซสชัน มีดังนี้

1) เมื่อผู้ใช้เชื่อมต่อไปยังเว็บไซต์แต่ละแห่ง เว็บเซิร์ฟเวอร์จะสร้างรหัสสำหรับการอ้างอิงถึงผู้ใช้แต่ละคน โดยรหัสนี้เรียกว่า Session ID หรือ SID ประกอบด้วยตัวเลข 0 - 9 และตัวอักษร a - z จำนวน 32 ตัว เช่น d77a4f704b938b240e4228a7e0847895 ทั้งนี้ค่า SID ของผู้ใช้แต่ละคนที่เชื่อมต่อกับเว็บไซต์ในช่วงเวลาเดียวกันจะต้องมี SID ไม่ซ้ำกัน

2) ค่า SID นี้จะถูกนำไปใช้ในการอ้างอิงถึงผู้ใช้คนนั้นตลอดการเชื่อมต่อ และเมื่อใดก็ตามที่ผู้ใช้เลิกการเชื่อมต่อ เช่น การปิดบราวเซอร์ จะส่งผลให้ค่า SID ของผู้ใช้คนนั้นถูกยกเลิกไป จนกว่าจะมีการเชื่อมต่อไปยังเว็บไซต์แห่งนั้น ครั้งต่อไปค่า SID จึงจะถูกสร้างขึ้นมาให้ใหม่เป็นเช่นนี้ไปตลอด

3) เมื่อสร้างข้อมูลแบบเซสชันสำหรับผู้ใช้คนใด ข้อมูลนั้นก็จะมีผลหรือใช้งานได้เฉพาะกับผู้ใช้คนนั้นส่วนคนอื่นๆ ไม่สามารถเรียกใช้งานได้ เนื่องจากข้อมูลเซสชันของผู้ใช้แต่ละคนจะถูกแยกออกจากกัน

4) ข้อมูลแบบเซสชันที่สร้างขึ้นจากเว็บเพจหนึ่งจะสามารถนำไปใช้งานที่เว็บเพจอื่นๆ ที่ถูกเรียกใช้โดยผู้ใช้ที่เป็นผู้สร้างเซสชันนั้นได้

5) ข้อมูลเซสชันที่ถูกสร้างขึ้นจะสามารถใช้งานได้ชั่วระยะเวลาที่เปิดใช้บราวเซอร์อยู่ แต่ไม่เกิน 180 นาที ถ้าปิดบราวเซอร์หรือเวลาเกินกว่า 180 นาที ข้อมูลก็จะถูกยกเลิกไป หรือนอกจากนี้ ยังสามารถสั่งลบข้อมูลเซสชันเมื่อไม่ต้องการใช้งานต่อไปได้

6) เซสชันจะมีความเกี่ยวข้องกับคุกกี้ โดยเซสชันจะใช้การเก็บข้อมูลรหัสแบบคุกกี้ไว้ที่เครื่องของผู้ใช้แต่ละคน และเมื่อเชื่อมต่อไปยังเว็บเซิร์ฟเวอร์ ข้อมูลนี้จะถูกนำไปใช้ในการตรวจสอบว่าเป็นผู้ใช้คนใด ดังนั้นหากเครื่องของผู้ใช้คนใดปฏิเสธการใช้คุกกี้ก็จะส่งผลถึงเซสชันด้วย แต่ปัจจุบันนี้มีการใช้เทคนิค URL Rewriting หรือนำรหัสเซสชันไปต่อท้าย URL ในแบบ Query String ในชื่อตัวแปร PHPSESSID

7) แม้เซสชันจะเป็นการพักเก็บข้อมูลไว้ชั่วคราวคล้ายกับคุกกี้ แต่ทั้งคุกกี้และเซสชันก็มีแนวทางการนำไปใช้งานที่แตกต่างกัน

10.4.2 การเริ่มการทำงานของเซสชัน

ก่อนที่จะเริ่มใช้งานเซสชัน ต้องเริ่มด้วยการสั่งให้เซสชันเริ่มทำงานด้วยฟังก์ชัน session_start ( ) โดยต้องกำหนดไว้ที่ส่วนบนสุดของเว็บเพจ ตัวอย่างการใช้งานดังนี้

ตัวอย่างที่ 10.13 การเริ่มการทำงานของเซสชัน

```
1    <?php
2        session_start ( );
3    ?>
```

จากตัวอย่างที่ 10.13 แสดงตัวอย่างการเริ่มการทำงานของเซสชัน โดยต้องกำหนดไว้ที่ส่วนบนสุดของเว็บเพจ

หมายเหตุ

ข้อควรระวัง คือ เนื่องจากเซสชันต้องอาศัยคุกกี้เป็นกลไกร่วมด้วย และคุกกี้จะอาศัยการส่งข้อมูลแบบเฮดเดอร์ ทำให้เซสชันมีความเกี่ยวพันกับเฮดเดอร์ด้วย ดังนั้นการใช้ฟังก์ชัน session_start ( ) ต้องทำก่อนการส่งข้อมูลใดๆ ไปที่บราวเซอร์เช่นเดียวกับเฮดเดอร์และคุกกี้

1) การเก็บข้อมูลด้วยตัวแปร $_SESSION

ตัวแปร $_SESSION เป็นตัวแปรอาร์เรย์สำหรับจัดเก็บข้อมูลแบบเซสชัน ตัวแปรนี้จะมีลักษณะเป็นคีย์และค่าอาเรย์ โดยที่คีย์ คือ ชื่อของเซสชัน (คล้ายกับชื่อของคุกกี้) และค่าอาร์เรย์ คือ ค่าของเซสชัน ทั้งนี้จะสร้างตัวแปร $_SESSION จำนวนเท่าไหร่ก็ได้ ตัวอย่างดังนี้

ตัวอย่างที่ 10.14 การเก็บข้อมูลด้วยตัวแปร $_SESSION

```
1    <?php
2        session_start ( );
3        $_SESSION ["user"] = "Parinya";
4        $_SESSION ["pass"] = $_POST ["pass"];
5    ?>
```

จากตัวอย่างที่ 10.14 แสดงการเก็บข้อมูลด้วยตัวแปร $_SESSION โดยเมื่อต้องการนำค่าเซสชันใดก็สามารถระบุผ่านตัวแปร $_SESSION พร้อมระบุชื่อเซสชันที่ต้องการ แต่อย่างไรก็ตาม เนื่องจากเซสชันสามารถถูกลบ หรือหมดอายุได้ ก่อนอ่านค่าตัวแปรควรตรวจสอบการตัวแปรเซสชันก่อนการเรียกใช้งาน โดยใช้ฟังก์ชัน isset ( ) ตรวจสอบก่อนเรียกใช้งานตัวแปร ตัวอย่างดังนี้

ตัวอย่างที่ 10.15 การใช้งานตัวแปร $_SESSION

```
1    <?php
2        session_start ( );
3        if (isset ($_SESSION ["login"])) $login = $_SESSION ["login"];
4    ?>
```

10.4.3 การลบข้อมูลเซสชัน

ปกติแล้วเมื่อบราวเซอร์ที่เปิดเอาไว้ถูกปิดทั้งหมด หรือระยะเวลาที่สร้างเซสชันเอาไว้นานเกินกว่า 180 นาที (3 ชั่วโมง) ค่าตัวแปรเซสชันที่ถูกสร้างขึ้นก็จะยกเลิกไปเองโดยอัตโนมัติ แต่บางครั้งก็ต้องการยกเลิกข้อมูลเซสชันที่เวลาใดเวลาหนึ่งขณะที่ยังใช้งานเว็บไซต์อยู่ เช่น กรณีการออกจากระบบ (Logout) เป็นต้น การลบข้อมูลเซสชันสามารถใช้ฟังก์ชัน ดังนี้

1) ฟังก์ชัน unset ( ) เป็นฟังก์ชันที่ใช้ในการยกเลิกตัวแปรทั่วๆ ไปของ PHP สามารถนำมาใช้กับตัวแปรเซสชันได้เช่นกัน มีรูปแบบดังนี้

รูปแบบ

```
void unset ( mixed $var [, mixed $... ] )
```

เมื่อ $var หมายถึง ชื่อของตัวแปรที่จะลบหรือยกเลิก

ตัวอย่างที่ 10.16 การใช้ฟังก์ชัน unset ( )

```
1   <?php
2       session_start ( );
3       $_SESSION ["name"] = "Mr.Parinya";
4       echo $_SESSION ["name"] . "<br>";
5       unset ($_SESSION ["name"]);
6       echo $_SESSION ["name"] . "<br>";       // จะไม่พบข้อมูลใดๆ เนื่องจากได้ลบแล้ว
7   ?>
```

2) ฟังก์ชัน session_destroy ( ) เป็นฟังก์ชันที่ใช้สำหรับลบข้อมูลหรือยกเลิกตัวแปรเซสชันทั้งหมดในระบบที่ได้สร้างขึ้นให้โดยผู้ใช้คนนั้นๆ มีรูปแบบดังนี้

รูปแบบ

```
bool session_destroy ( void )
```

หมายเหตุ

ฟังก์ชัน session_destroy ( ) ใช้สำหรับลบหรือยกเลิกตัวแปรเซสชันทั้งหมด หากต้องการลบตัวแปรเซสชันตัวใดตัวหนึ่ง ขอแนะนำให้เลือกใช้ฟังก์ชัน unset ( )

10.4.4 ฟังก์ชันอื่นๆ ที่เกี่ยวข้องกับเซสชัน

1) ฟังก์ชัน session_id ( ) เป็นฟังก์ชันที่ใช้สำหรับคืนค่า SID มีรูปแบบดังนี้

รูปแบบ

```
string session_id ([ string $id ])
```

ตัวอย่างที่ 10.17 การใช้ฟังก์ชัน session_id ( )

```
1    <?php
2        session_start ( );
3        echo session_id ( );
4    ?>
```

2) ฟังก์ชัน session_regenerate_id ( ) เป็นฟังก์ชันที่ใช้สำหรับการสร้าง SID ใหม่

รูปแบบ

```
bool session_regenerate_id ([ bool $delete_old_session = false ] )
```

ตัวอย่างที่ 10.18 การใช้ฟังก์ชัน session_regenerate_id ( )

```
1    <?php
2        session_start ( );
3        echo session_id ( );
4        session_regenerate_id ( );
5        echo session_id ( );
6    ?>
```

3) ฟังก์ชัน session_encode ( ) เป็นฟังก์ชันที่ใช้สำหรับเข้ารหัสข้อมูลเซสชันทั้งหมด โดยจะคืนค่าข้อมูลที่เข้ารหัสไว้แล้ว

รูปแบบ

```
string session_encode ( void )
```

ตัวอย่างที่ 10.19 การใช้ฟังก์ชัน session_encode ( )

```
1    <?php
2        session_start ( );
3        $_SESSION ["user"] = "parinya";
4        $_SESSION ["pass"] = "1234";
5        echo session_encode ( );
6    ?>
```

4) ฟังก์ชัน session_decode ( ) เป็นฟังก์ชันที่ใช้สำหรับถอดรหัสข้อมูล ที่ถูกเข้ารหัสโดย session_encode ( ); มีรูปแบบดังนี้

รูปแบบ

```
bool session_decode ( string $data )
```

ตัวอย่างที่ 10.20 การใช้ฟังก์ชัน session_decode ( )

```
1    <?php
2        session_start ( );
3        $_SESSION["user"] = "parinya";
4        $_SESSION["pass"] = "1234";
5        $secret = session_encode ( );
6        session_decode ( $secret );
7    ?>
```

5) ฟังก์ชัน session_register ( ) เป็นฟังก์ชันที่ใช้สำหรับใช้ลงทะเบียน ตัวแปร ให้เป็นตัวแปรเซสชันมีรูปแบบดังนี้

รูปแบบ

```
bool session_register ( mixed $name [, mixed $... ] )
```

ตัวอย่างที่ 10.21 การใช้ฟังก์ชัน session_register ( )

```
1    <?php
2        session_start ( );
3        $the_best = "Computer Science of SRU";
4        session_register ("the_best");
5        echo $_SESSION["the_best"];
6    ?>
```

จากตัวอย่างที่ 10.21 แสดงการใช้ฟังก์ชัน session_register ( ) เพื่อลงทะเบียนตัวแปรเซสชัน อธิบายดังนี้

บรรทัดที่ 2 เริ่มต้นใช้งานเซสชัน

บรรทัดที่ 3 กำหนดให้ตัวแปร $the_best มีค่าเท่ากับ "Computer Science of SRU"

บรรทัดที่ 4 ลงทะเบียนตัวแปร $the_best เป็นตัวแปรชนิดเซสชัน

บรรทัดที่ 5 แสดงผลตัวแปรเซสชัน$_SESSION["the_best"] ที่ได้ลงทะเบียนไว้ในบรรทัดที่ 4

6) ฟังก์ชัน session_unregister ( ) เป็นฟังก์ชันที่ใช้สำหรับยกเลิกการลงทะเบียนตัวแปรเซสชันมีรูปแบบดังนี้

รูปแบบ

```
bool session_register ( mixed $name [, mixed $... ] )
```

ตัวอย่างที่ 10.22 การใช้ฟังก์ชัน session_unregister ( )

```
1    <?php
2        $the_best = "Computer Science of SRU";
3        session_register ("the_best");
4        echo $_SESSION ["the_best"];
5        session_unregister ("the_best");
6        echo $_SESSION ["the_best"];
7    ?>
```

จากตัวอย่างที่ 10.22 การใช้ฟังก์ชัน session_unregister ( ) สำหรับยกเลิกการลงทะเบียนตัวแปรเซสชัน อธิบายดังนี้

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $the_best มีค่าเท่ากับ "Computer Science of SRU"

บรรทัดที่ 3 ลงทะเบียนตัวแปรเซสชัน session_register ("the_best")

บรรทัดที่ 4 แสดงผลค่าตัวแปรเซสชัน จะได้ผลลัพธ์ คือ "Computer Science of SRU"

บรรทัดที่ 5 ยกเลิกตัวแปรเซสชัน

บรรทัดที่ 6 แสดงผลค่าตัวแปรเซสชัน (ไม่แสดงข้อมูลใดๆ เนื่องจากตัวแปรเซสชันได้ถูกยกเลิกแล้ว)

7) ฟังก์ชัน session_unset ( ) เป็นฟังก์ชันที่ใช้สำหรับใช้ยกเลิกตัวแปรเซสชันทั้งหมด คือ ยกเลิกตัวแปร ที่ได้ลงทะเบียนไว้มีรูปแบบดังนี้

รูปแบบ

```
void session_unset ( void )
```

ตัวอย่างที่ 10.23 การใช้ฟังก์ชัน session_unset ( )

```
1    <?php
2        session_start ( );
3        $the_best = "Computer Science of SRU";
4        session_register ( "the_best" );
5        session_unset ( );
6    ?>
```

8) ฟังก์ชัน session_is_registered ( ) เป็นฟังก์ชันที่ใช้สำหรับใช้ตรวจสอบว่าตัวแปรนั้นได้ลงทะเบียนเซสชันแล้วหรือยังมีรูปแบบดังนี้

รูปแบบ

```
bool session_is_registered ( string $name )
```

ตัวอย่างที่ 10.24 การใช้งานฟังก์ชัน session_is_registered ( )

```
1    <?php
2        session_start ( );
3        $the_best = "Computer Science of SRU";
4        if ( ! session_is_registered ( $the_best ) ) {
5            session_register( $the_best );
6        }
7    ?>
```

จากตัวอย่างที่ 10.24 ตัวอย่างการใช้งานฟังก์ชัน session_is_registered ( ) อธิบายดังนี้

บรรทัดที่ 2 เริ่มต้นใช้งานเซสชัน

บรรทัดที่ 3 กำหนดให้ตัวแปร $the_best มีค่าเท่ากับ "Computer Science of SRU"

บรรทัดที่ 4 ใช้โครงสร้างเงื่อนไข if ร่วมกับฟังก์ชัน session_is_registered ( ) เพื่อตรวจสอบตัวแปรที่ระบุได้ลงทะเบียนเซสชันแล้วหรือไม่ หากยังไม่ได้ลงทะเบียนไปบรรทัด 5

บรรทัดที่ 5 ลงทะเบียนตัวแปร $the_best เป็นตัวแปรชนิดเซสชัน

10.5 ขั้นตอนการประยุกต์ใช้เซสชัน เพื่อเป็นแนวทางในการพัฒนาระบบ Login

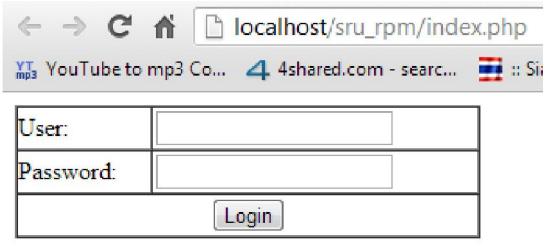
10.5.1 สร้างฟอร์มสำหรับการ Login ประกอบด้วย User Name และ Password ดังตัวอย่างที่ 10.25

ตัวอย่างที่ 10.25 ฟอร์มสำหรับ Login (index.php) มีสคริปต์ ดังนี้

```
<?php
    session_start ( );
    session_destroy ( );
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ตัวอย่างหน้า Login</title>
</head>
<body>
```

```
        <form id="form1" name="frmLogin" method="POST" action="check_login.php">
                <table width="300" border="1" cellspacing="0" cellpadding="0">
                <tr>
                        <td width="117">User:</td>
                        <td width="283"><label>
                        <input type="text" name="user" id="user" />
                        </label></td>
                </tr>
                <tr>
                        <td>Password:</td>
                        <td><label>
                        <input type="password" name="pass" id="pass" />
                        </label></td>
                </tr>
                <tr>
                        <td colspan="2" align="center"><label>
                        <input type="submit" name="login" id="login" value="Login" />
                        </label></td>
                </tr>
                </table>
        </form>
</body>
</html>
```

จากตัวอย่างที่ 10.25 ฟอร์มสำหรับ Login เพื่อเข้าสู่ระบบ กำหนดส่วนนำเข้าข้อมูลที่สำคัญ 3 ส่วน คือ 1) ส่วนนำเข้าข้อมูล ประกอบด้วย user และ pass 2) รูปแบบวิธีส่งฟอร์มใช้รูปแบบวิธีแบบ POST และ 3) สคริปต์ที่จะประมวลผลต่อ คือ check_login.php ผลลัพธ์ที่ได้ ดังภาพที่ 10.2



ภาพที่ 10.2 สร้างฟอร์มสำหรับการ Login กรณีศึกษาและประยุกต์ใช้งานเซสชัน

10.5.2 สร้างไฟล์เพื่อตรวจสอบข้อมูลที่ผู้ใช้ Login ผ่านส่วนนำเข้าข้อมูล user และ pass กำหนดให้ใช้ชื่อ check_login.php มีสคริปต์คำสั่ง ดังนี้

ตัวอย่างที่ 10.26 สคริปต์สำหรับตรวจสอบข้อมูลที่ผู้ใช้ Login เข้ามา (check_login.php)

```
1   <?php
2       session_start ( );
3       $user_check="testuser";
4       $pass_check="testpass";
5       if (($_POST['user']==$user_check) && ($_POST['pass']==$pass_check)) {
6           $user = $_POST['user'];
7           $pass = $_POST['pass'];
8           $fullname = "Mr.Parinya Noidonprai";
9           session_register ("fullname");
10          session_register ("user");
11          session_register ("pass");
12          header ("location: member.php");
13      } else header ("location: index.php");
14  ?>
```

จากตัวอย่างที่ 10.26 สคริปต์สำหรับตรวจสอบข้อมูลที่ผู้ใช้ Login เข้ามา อธิบายดังนี้

บรรทัดที่ 2 เริ่มใช้งานเซสชัน

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร $user_check มีค่าเท่ากับ testuser (ใช้เปรียบเทียบชื่อผู้ใช้กับส่วนนำเข้าข้อมูลที่ผู้ใช้งานป้อนผ่าน user)

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร $pass_check มีค่าเท่ากับ testpass (ใช้เปรียบเทียบรหัสผ่านกับส่วนนำเข้าข้อมูลที่ผู้ใช้งานป้อนผ่าน pass)

บรรทัดที่ 5 ใช้โครงสร้างเงื่อนไข if ตรวจสอบส่วนนำเข้าที่รับเข้ามากับค่าที่กำหนดไว้ตรงกันหรือไม่ ถ้าชื่อผู้ใช้และรหัสผ่านเหมือนกับค่าที่กำหนดไว้ในตัวแกร บรรทัดที่ 3 และ 4 ไปบรรทัดที่ 6 ถ้าไม่ใช่ไปทำคำสั่งในบรรทัดที่ 13

บรรทัดที่ 6 กำหนดให้ตัวแปร $user มีค่าเท่ากับส่วนนำเข้าข้อมูล $_POST['user']

บรรทัดที่ 7 กำหนดให้ตัวแปร $pass มีค่าเท่ากับส่วนนำเข้าข้อมูล $_POST['pass']

บรรทัดที่ 8 กำหนดตัวแปรอื่นๆ ตามผู้ใช้จะใช้งานเซสชัน จากตัวอย่างกำหนดให้ตัวแปร $fullname มีค่าเท่ากับ "Mr.Parinya Noidonprai"

บรรทัดที่ 9 ลงทะเบียนตัวแปร fullname ให้เป็นตัวแปรเซสชัน

บรรทัดที่ 10 ลงทะเบียนตัวแปร user ให้เป็นตัวแปรเซสชัน

บรรทัดที่ 11 ลงทะเบียนตัวแปร pass ให้เป็นตัวแปรเซสชัน

บรรทัดที่ 12 ใช้ฟังก์ชัน header ( ) เพื่อย้ายหน้าเว็บเพจไปยัง member.php

บรรทัดที่ 13 โครงสร้างเงื่อนไข else (กรณีที่ชื่อผู้ใช้กับรหัสผ่านที่ป้อนผ่านส่วนนำเข้าข้อมูลไม่ตรงกับค่าตัวแปรที่กำหนดไว้) ใช้ฟังก์ชัน header ( ) เพื่อกลับไปยังเว็บเพจ index.php (กลับไป Login ใหม่)

10.5.3 กำหนดหน้าเว็บเพจที่จะประมวลผลต่อ เมื่อตรวจสอบชื่อผู้ใช้และรหัสผ่านถูกต้องให้ทำงานต่อที่เว็บเพจ member.php มีสคริปต์คำสั่ง ดังนี้

ตัวอย่างที่ 10.27 สคริปต์ตรวจสอบการลงทะเบียนตัวแปรเซสชัน (member.php)

```
1       <?php
2       session_start ( );
3       if (isset ($_SESSION["user"]) && isset ($_SESSION["pass"]))
4               echo "Congratulation, You are login with: ". $_SESSION["fullname"];
5       else  header ("location: index.php");
6       ?>
```

จากภาพที่ 10.27 สคริปต์ตรวจสอบการลงทะเบียนตัวแปรเซสชัน อธิบายดังนี้

บรรทัดที่ 2 เริ่มต้นใช้งานเซสชัน

บรรทัดที่ 3 ตรวจสอบการลงทะเบียนตัวแปรเซสชัน $_SESSION["user"] และ $_SESSION["pass"] หากไม่มีการลงทะเบียนตัวแปรดังกล่าวแสดงว่าเข้าสู่สคริปต์นี้อย่างไม่ถูกต้อง ไปทำงานต่อบรรทัดที่ 5 กรณีตรวจสอบแล้วพบการลงทะเบียนตัวแปรเซสชันที่ระบุ ไปบรรทัดที่ 4

บรรทัดที่ 4 แสดงความยินดีต้อนรับ หรือผู้พัฒนาระบบสามารถประยุกต์ใช้ฟังก์ชัน header ( ) เพื่อย้ายหน้าเว็บเพจไปยังหน้าเว็บเพจของระบบสมาชิก หรือเว็บเพจอื่นๆ ตามต้องการ

บรรทัดที่ 5 กรณีที่พบตัวแปรเซสชันที่ลงทะเบียนกลับไป Login ใหม่ที่ index.php

สรุป

ลักษณะของเว็บแอพพลิเคชันที่พัฒนาเป็นระบบงานนั้น จำเป็นต้องแบ่งเว็บเพจออกเป็นส่วนๆ แยกตามบทบาทหน้าที่การทำงาน เช่น ตรวจสอบสิทธิผู้ใช้งานระบบ จัดการข้อมูลสมาชิก สืบค้นข้อมูล รายงาน แก้ไข และลบข้อมูล เป็นต้น และเมื่อเริ่มทำงานติดต่อกับผู้ใช้งานระบบ ทุกเว็บเพจภายในเว็บไซต์จำเป็นต้องทำงานประสานกันเป็นระบบ ดังนั้นจึงจำเป็นต้องเรียนรู้วิธีการควบคุมหน้าเว็บเพจ เช่น ย้ายไปยังเว็บเพจที่ทำหน้าที่ในเรื่องนั้นๆ ตามหน้าที่ที่ผู้พัฒนาได้ออกแบบไว้ เป็นต้น ปัญหาที่สำคัญเมื่อมีการย้ายหรือปรับเปลี่ยนเว็บเพจ คือ ตัวแปรที่ทำงานอยู่ภายในเว็บเพจนั้นๆ จะถูกยกเลิกการใช้งานโดยปริยายไม่สามารถนำไปใช้ในเว็บเพจอื่นๆ ได้ นี่คือสาเหตุที่ต้องเรียนรู้ถึงวิธีการส่งค่าตัวแปรในรูปแบบที่ได้กล่าวถึงในบทนี้

คำถามท้ายบท

1. จงอธิบายหน้าที่ของฟังก์ชัน header ( ) พร้อมยกตัวอย่างประกอบ

2. จงอธิบายหลักการส่งข้อมูลระหว่างเว็บเพจแบบ Query String และอธิบายถึงหลักการอ่านข้อมูลที่เว็บเพจปลายทางที่ได้รับข้อมูล Query String

3. จงอธิบายถึงประโยชน์ของคุกกี้และเซสชัน และความแตกต่างระหว่างคุกกี้และเซสชัน

4. จงอธิบายหน้าที่ของฟังก์ชัน session_start ( ) และฟังก์ชัน session_destroy ( ) พร้อมยกตัวอย่างสคริปต์ประกอบการอธิบาย

5. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากสคริปต์

```
1    <?php
2        session_start ( );
3        $the_best = "Computer Science of SRU";
4        session_register ("the_best");
5        echo $_SESSION["the_best"];
6    ?>
```