บทที่ 9
การจัดการข้อมูลจากฟอร์ม

กระบวนการที่สำคัญของการสร้างเว็บแอพพลิเคชัน คือ การนำข้อมูลที่ถูกส่งจากเว็บบราวเซอร์ไปประมวลผลในฝั่งของเว็บเซิร์ฟเวอร์ จึงทำให้ต้องมีส่วนในการกำหนดข้อมูลที่จะส่งไปประมวลผล ดังนั้นในบทนี้จะศึกษาเกี่ยวกับการสร้างส่วนสำหรับรับข้อมูลจากผู้ใช้งาน เรียกว่า "ฟอร์ม (Form)" โดยภายในฟอร์มนี้จะประกอบด้วยการรับข้อมูลได้หลายชนิด การรับข้อมูลในรูปแบบที่แตกต่างกัน และขั้นตอนทางด้านเซิร์ฟเวอร์ก็จะกำหนดรูปแบบวิธีการรับข้อมูลจากฟอร์ม จะแตกต่างตามชนิดของข้อมูลที่รับเข้าไปประมวลผล ในบทนี้จะอธิบายการสร้างฟอร์มและกำหนดส่วนที่ใช้รับข้อมูลต่างๆ รวมถึงวิธีการส่งข้อมูลเพื่อนำข้อมูลที่รับเข้ามาไปประมวลผลหรือจัดเก็บข้อมูลลงในไฟล์หรือฐานข้อมูลต่อไป ซึ่งถือเป็นสิ่งสำคัญอย่างยิ่งสำหรับการสร้างเว็บแอพพลิเคชัน

9.1 กำหนดขอบเขตของฟอร์ม

ก่อนจะเริ่มสร้างรับข้อมูลต่างๆ จะต้องกำหนดขอบเขตให้กับตัวฟอร์ม โดยการกำหนดขอบเขตฟอร์ม ด้วยการแทรกแท็ก <form> ... </form> เข้าไปในเว็บเพจ การใช้งานแท็ก <form> มีรูปแบบการใช้งานและคุณสมบัติที่จำเป็น ดังนี้

รูปแบบ

```
<form name = "ชื่อฟอร์ม" method = "วิธีส่งข้อมูล" action = "ชื่อไฟล์สคริปต์ที่จะประมวลผล">
...
</form>
```

โดยวิธีการส่งข้อมูลมีอยู่ด้วยกัน 2 รูปแบบ ประกอบด้วย POST และ GET ทั้ง 2 รูปแบบมีลักษณะการใช้งานที่แตกต่างกันทั้งนี้การเลือกวิธีการส่งข้อมูลจะขึ้นอยู่กับความเหมาะสมของงาน ดังนี้

ตารางที่ 9.1 แสดงลักษณะการใช้งานวิธีการส่งข้อมูลแบบ POST และ GET

วิธีส่งข้อมูล	ลักษณะการใช้งาน
POST	เป็นการส่งข้อมูลจากฟอร์มไปยังสคริปต์โดยตรง ซึ่งข้อมูลที่ส่งผ่านไม่สามารถมองเห็นได้ด้วยวิธีปกติ วิธีนี้จึงเหมาะสำหรับข้อมูลที่ต้องการปิดเป็นความลับ
GET	เป็นการส่งข้อมูลจากฟอร์มไปยังสคริปต์โดยผ่าน URL ของเว็บเพจ ซึ่งวิธีนี้สามารถมองเห็นข้อมูลที่ส่งผ่านในช่อง Address Bar ของเว็บบราวเซอร์ ดังนั้นจึงไม่เหมาะกับข้อมูลที่เป็นความลับ

9.2 การใช้งานส่วนนำเข้าข้อมูล

การสร้างฟอร์มขึ้นมาหนึ่งฟอร์ม ไม่ว่าจะเป็นฟอร์มสมัครสมาชิก ฟอร์มสั่งซื้อสินค้า หรือฟอร์มสำหรับล็อกอินเข้าสู่ระบบสมาชิก จะประกอบไปด้วยส่วนนำเข้าข้อมูลหลากหลายรูปแบบ เพื่อให้ง่ายแก่การทำความเข้าใจ ดังนั้นจะแบ่งส่วนนำเข้าข้อมูล ออกเป็น 4 กลุ่ม ประกอบด้วย 1) กลุ่มช่องรับข้อความ 2) กลุ่มตัวเลือก 3) กลุ่มปุ่มส่งข้อมูล และ 4) กลุ่มอื่นๆ ทั้งนี้การประยุกต์ใช้งานในแต่ละกลุ่มขึ้นอยู่กับความเหมาะสมของข้อมูลในฟอร์มและความต้องการ โดยแต่ละกลุ่มมีรายละเอียด ดังนี้

9.2.1 กลุ่มช่องรับข้อความ

กลุ่มช่องรับข้อความจะเป็นกลุ่มที่นิยมใช้งานมากที่สุด สามารถกำหนดช่องรับข้อความได้ 3 รูปแบบ คือ 1) ช่องรับข้อความบรรทัดเดียว (Single Line) 2) ช่องรับข้อความหลายบรรทัด (Multi Line) และ 3) ข้อความรหัสผ่าน (Password) รายละเอียดดังนี้

1) ช่องรับข้อความบรรทัดเดียว เหมาะสำหรับการป้อนข้อมูลสั้นๆ เช่น ชื่อ-สกุล อีเมล์ เป็นต้น สามารถเขียนสคริปต์และคุณสมบัติให้กับช่องรับข้อความแบบบรรทัดเดียว โดยกำหนดผ่านแท็ก <input> มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
<input type = "text" name = "ชื่อส่วนนำเข้าข้อมูล" size = "ความกว้างของช่องรับข้อความ" maxlength = "จำนวนตัวอักษรสูงสุดที่สามารถพิมพ์ลงในช่องได้">
```

สำหรับคุณสมบัติ size และ maxlength จะกำหนดหรือไม่ก็ได้

การตั้งชื่อช่องรับข้อความควรสื่อความหมายของข้อมูลในช่องนั้นๆ ด้วยชื่อสั้นๆ จดจำง่าย และที่สำคัญที่สุดจะต้องตั้งชื่อให้กับส่วนนำเข้าข้อมูลทุกตัว และชื่อที่ตั้งจะต้องไม่ซ้ำกัน เนื่องจากชื่อของส่วนนำเข้าข้อมูลเหล่านี้จะใช้ในการแทนค่าช่องรับข้อมูลในฟอร์มเพื่อนำไปอ้างอิง เรียกใช้ และประมวลผลต่อไป

2) ช่องรับข้อความหลายบรรทัด เหมาะสำหรับข้อมูลที่มีเนื้อหายาวๆ หลายบรรทัด เช่น ข้อมูลความคิดเห็น ข้อมูลที่อยู่ เป็นต้น สามารถเขียนสคริปต์และคุณสมบัติให้กับช่องรับข้อความแบบหลายบรรทัด โดยกำหนดผ่านแท็ก <textarea> ... </textarea> มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
<textarea name = "ชื่อส่วนนำเข้าข้อมูล" cols = " จำนวนตัวอักษรที่สามารถพิมพ์ได้ใน 1 บรรทัด" row = " จำนวนบรรทัดที่แสดงผล">
```

3) ช่องรับข้อความรหัสผ่าน ส่วนนำเข้าข้อมูลชนิดนี้จะมีความพิเศษกว่าช่องรับข้อความที่ผ่านมา ซึ่งเมื่อผู้ใช้ป้อนข้อมูลลงในช่องรับข้อความรหัสผ่าน ข้อมูลจะกลายเป็นเครื่องหมายวงกลมดำแทนตัวอักษร เพื่อป้องกันไม่ให้ผู้อื่นเห็นข้อมูลรหัสผ่าน สามารถเขียนสคริปต์และคุณสมบัติให้กับช่องรับข้อความรหัสผ่าน โดยกำหนดผ่านแท็ก <input> มีรูปแบบการใช้งานดังนี้

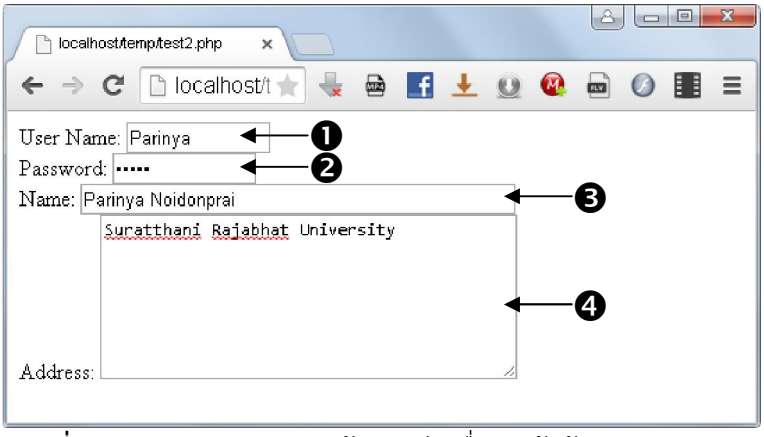
รูปแบบ

```
<input type = "password" name = "ชื่อส่วนนำเข้าข้อมูล" size = "ความกว้างของช่องรับข้อความ" maxlength = "จำนวนตัวอักษรสูงสุดที่สามารถพิมพ์ลงในช่องได้">
```

ตัวอย่างที่ 9.1 สคริปต์การสร้างฟอร์มรับข้อมูลแบบช่องรับข้อความ

```
<form name= "frmMember" method = "POST" action = "member.php">
    User Name: <input name = "txtUser" type = "text" size = "10"><br>
    Password: <input name = "txtPass" type = "password" size = "10"><br>
    Name: <input name = "txtName" type = "text" size = "40"><br>
    Address: <textarea name = "txtAddress" cols = "40" row = "5"></textarea>
</form>
```

ผลลัพธ์ แสดงดังภาพที่ 9.1



ภาพที่ 9.1 แสดงตัวอย่างการสร้างฟอร์มเพื่อนำเข้าข้อมูล

จากภาพที่ 9.1 แสดงตัวอย่างการสร้างฟอร์มรับข้อมูล ประกอบด้วยช่องรับข้อความดังนี้

❶ ช่องรับข้อความบรรทัดเดียว ชื่อส่วนนำเข้าข้อมูล คือ txtUser

❷ ช่องรับข้อความรหัสผ่าน ชื่อส่วนนำเข้าข้อมูล คือ txtPass

❸ ช่องรับข้อความบรรทัดเดียว ชื่อส่วนนำเข้าข้อมูล คือ txtName

❹ ช่องรับข้อความหลายบรรทัด ชื่อส่วนนำเข้าข้อมูล คือ txtAddress

9.2.2 กลุ่มตัวเลือก

กลุ่มตัวเลือก คือ กลุ่มที่ใช้กำหนดตัวเลือกให้กับผู้ใช้งานระบบสามารถคลิกเพื่อเลือกคำตอบที่ต้องการ สามารถสรุปได้ 2 ประเภท คือ 1) ตัวเลือกที่เลือกตอบได้เพียงข้อเดียว และ 2) ตัวเลือกที่เลือกตอบได้หลายข้อ ซึ่งแต่ละประเภทจะประกอบไปด้วยส่วนนำเข้าข้อมูลต่างๆ ดังนี้

1) ตัวเลือกที่เลือกตอบได้เพียงข้อเดียว มี 2 ชนิดย่อย คือ 1) ปุ่มตัวเลือกแบบปุ่มกลม (Radio Button) และ 2) รายการตัวเลือกแบบ Menu รายละเอียดดังนี้

1.1) ปุ่มตัวเลือกแบบปุ่มกลม ส่วนนำเข้าข้อมูลลักษณะปุ่มกลม ใช้สำหรับเลือกตัวเลือกได้เพียง 1 ตัวเลือกเท่านั้น ลักษณะการประยุกต์เลือกตอบ เช่น เพศ ศาสนา หรือสถานภาพสมรส เป็นต้น สามารถเขียนสคริปต์และคุณสมบัติให้กับปุ่มตัวเลือกแบบปุ่มกลม โดยกำหนดผ่านแท็ก <input> มีรูปแบบการใช้งานดังนี้

รูปแบบ

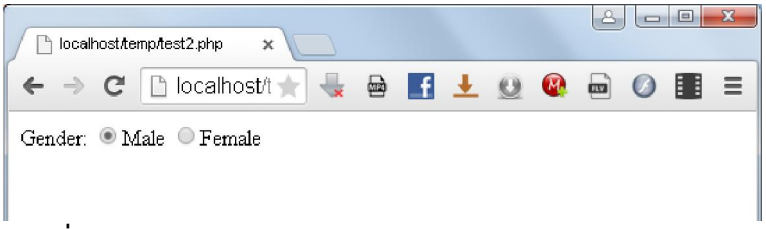
```
<input type = "radio" name = "ชื่อส่วนนำเข้าข้อมูล" value = "ค่าของตัวเลือก" checked = "checked">ข้อความแสดงเพื่อเลือกตอบ</input>
```

สำหรับคุณสมบัติ checked จะกำหนดหรือไม่ก็ได้ โดยจะใช้ในกรณีที่ต้องการกำหนดให้ตัวเลือกนี้ถูกเลือกไว้เป็นค่าเริ่มต้น ตัวอย่างดังนี้

ตัวอย่างที่ 9.2 สคริปต์การสร้างฟอร์มรับข้อมูลตัวเลือกแบบปุ่มกลม

```
<form name= "frmGender" method = "POST" action = "gender.php">
    Gender:
    <input type ="radio" name = "rdoGender" value = "M" checked = "checked">Male
    <input type ="radio" name = "rdoGender" value = "F">Female
</form>
```

ผลลัพธ์



ภาพที่ 9.2 แสดงตัวอย่างการใช้งานปุ่มตัวเลือกแบบปุ่มกลม

จากภาพที่ 9.2 แสดงตัวอย่างการใช้งานปุ่มตัวเลือกแบบปุ่มกลม สำหรับเลือกเพศ โดยกำหนดให้ชื่อกลุ่มตัวเลือกมีชื่อว่า rdoGender และกำหนดให้เลือกข้อมูล Male (เพศชาย) เป็นค่าเริ่มต้น

1.2) รายการตัวเลือกแบบ Menu จะมีลักษณะเป็นกล่องรายการแสดงขึ้นมาให้เห็นเพียงตัวเลือกเดียว เพื่อให้คลิกเลือกรายการ สามารถเขียนสคริปต์และคุณสมบัติให้กับรายการตัวเลือกแบบ Menu โดยกำหนดผ่านแท็ก <select> มีรูปแบบการใช้งานดังนี้

รูปแบบ

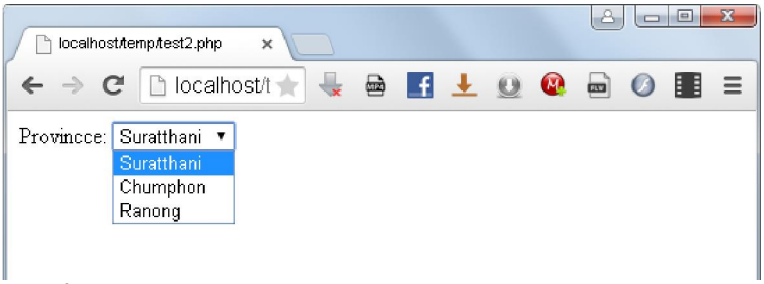
```
<select name = "ชื่อส่วนนำเข้าข้อมูล">
        <option value = "ค่าของตัวเลือกที่ 1" selected = "selected">ตัวเลือกที่ 1 </option>
        <option value = "ค่าของตัวเลือกที่ 2">ตัวเลือกที่ 2 </option>
```

```
        <option value = "ค่าของตัวเลือกที่ 2">ตัวเลือกที่ N </option>
    </select>
```

ตัวอย่างที่ 9.3 สคริปต์การสร้างฟอร์มรับข้อมูลรายการตัวเลือกแบบ Menu

```
<form name= "frmProvince" method = "POST" action = "province.php">
    Provincce:
    <select name = "sltProvince">
        <option value = "1" selected = "selected">Suratthani</option>
        <option value = "2">Chumphon</option>
        <option value = "3">Ranong</option>
    </select>
</form>
```

ผลลัพธ์



ภาพที่ 9.3 แสดงตัวอย่างการใช้งานรายการตัวเลือกแบบ Menu

จากภาพที่ 9.3 แสดงตัวอย่างการใช้งานรายการตัวเลือกแบบ Menu สำหรับเลือกจังหวัด โดยกำหนดให้ชื่อรายการตัวเลือกแบบ Menu มีชื่อว่า sltProvince และกำหนดให้เลือกข้อมูล Suratthani (จังหวัดสุราษฎร์ธานี) เป็นค่าเริ่มต้น

2) ตัวเลือกที่เลือกตอบได้หลายข้อ มี 1 ชนิด คือ ปุ่มตัวเลือกแบบปุ่มเหลี่ยม (Checkbox) โดยส่วนนำเข้าข้อมูล จะต้องมีชื่อกลุ่มตัวเลือกเดียวกันจึงจะสามารถเลือกตัวเลือกได้หลายตัวเลือก ข้อมูลที่ใช้ส่วนใหญ่จะเป็นข้อมูลเกี่ยวกับความชอบ สามารถเขียนสคริปต์และคุณสมบัติให้กับปุ่มตัวเลือกแบบปุ่มเหลี่ยม โดยกำหนดผ่านแท็ก <input> มีรูปแบบดังนี้

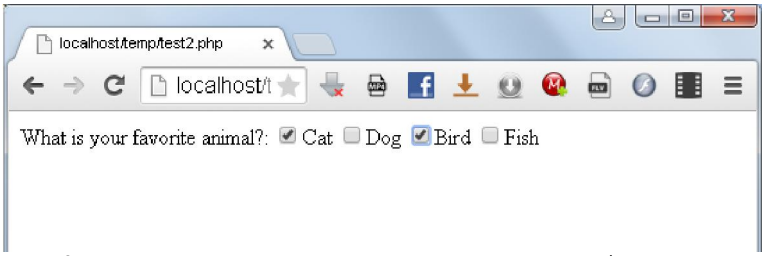
รูปแบบ

```
    <input type = "checkbox" name = "ชื่อส่วนนำเข้าข้อมูล" value = "ค่าของตัวเลือก"
checked = "checked">ข้อความให้เลือก</input>
```

ตัวอย่างที่ 9.4 สคริปต์การสร้างฟอร์มรับข้อมูลตัวเลือกแบบปุ่มเหลี่ยม (เลือกตอบได้มากกว่า 1)

```
<form name= "frmAnimal" method = "POST" action = "animal.php">
    What is your favorite animal?:
    <input type ="checkbox" name = "chkAni" value = "cat" checked = "checked">Cat
    <input type ="checkbox" name = "chkAni" value = "dog">Dog
    <input type ="checkbox" name = "chkAni" value = "bird">Bird
    <input type ="checkbox" name = "chkAni" value = "fish">Fish
</form>
```

ผลลัพธ์



ภาพที่ 9.4 แสดงตัวอย่างการใช้งานตัวเลือกแบบปุ่มเหลี่ยม

จากภาพที่ 9.4 แสดงตัวอย่างการใช้งานตัวเลือกแบบปุ่มเหลี่ยม สำหรับเลือกตอบรายการสัตว์เลี้ยงที่ชื่นชอบหรือประทับใจ สามารถเลือกตอบได้มากกว่า 1 รายการ โดยกำหนดให้กลุ่มตัวเลือกแบบปุ่มเหลี่ยมมีชื่อส่วนนำเข้าข้อมูล คือ chkAni และกำหนดให้เลือกข้อมูล Cat (แมว) เป็นค่าเริ่มต้น

9.2.3 กลุ่มปุ่มสั่งงาน

สำหรับส่วนนำเข้าข้อมูลในกลุ่มนี้มีความสำคัญมาก และพบเห็นมากที่สุด เนื่องจากเมื่อสร้างฟอร์มพร้อมส่วนนำเข้าข้อมูลที่เกี่ยวข้องทั้งหมดแล้ว หากขาดกลุ่มปุ่มสั่งงานจะทำให้ไม่สามารถส่งข้อมูลที่ผู้ใช้ป้อนไปยังสคริปต์ที่กำหนดไว้ในฟอร์มได้ กลุ่มปุ่มสั่งงานแบ่งออกเป็น 3 รูปแบบ ประกอบด้วย 1) ปุ่มส่งข้อมูล (Submit Button) 2) ปุ่มล้างข้อมูล (Reset Button) และ 3) ปุ่มส่งข้อมูลแบบรูปภาพ (Image Button) มีรายละเอียดดังนี้

1) ปุ่มส่งข้อมูล จะทำหน้าที่ส่งข้อมูลที่ผู้ใช้ป้อนไว้ในฟอร์มไปประมวลผลยังสคริปต์ที่ระบุไว้ในฟอร์ม สามารถเขียนสคริปต์และคุณสมบัติ โดยกำหนดผ่านแท็ก <input> มีรูปแบบดังนี้

รูปแบบ

```
<input type = "submit" name = "ชื่อปุ่ม" value = "ข้อความที่แสดงบนปุ่ม">
```

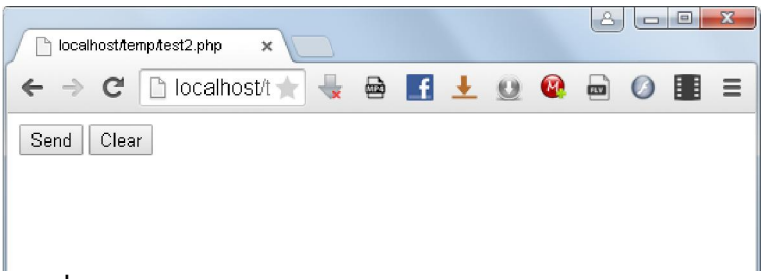
2) ปุ่มล้างข้อมูล จะทำหน้าที่ลบค่าข้อมูลที่ผู้ใช้ป้อนไว้ในฟอร์มขณะนั้นออกไปโดยไม่มีการส่งข้อมูลไปประมวลผล สามารถเขียนสคริปต์และคุณสมบัติ โดยกำหนดผ่านแท็ก <input> มีรูปแบบดังนี้

```
<input type = "reset" name = "ชื่อปุ่ม" value = "ข้อความที่แสดงบนปุ่ม">
```

ตัวอย่างที่ 9.5 แสดงตัวอย่างการใช้ปุ่มส่งข้อมูล และล้างข้อมูล

```
<form name= "frmButton" method = "POST" action = "test-button.php">
    <input type ="submit" name = "btnSubmit" value = "Send"><br>
    <input type ="reset" name = "btnReset" value = "Clear">
</form>
```

ผลลัพธ์



ภาพที่ 9.5 แสดงตัวอย่างการใช้งานปุ่มส่งค่าข้อมูล และล้างข้อมูล

3) ปุ่มส่งข้อมูลแบบรูปภาพ คือ การนำรูปภาพมาใช้แทนข้อความที่ปรากฏบนปุ่ม ทำให้เกิดความสวยงามหรือสร้างความโดดเด่นให้กับปุ่ม สามารถเขียนสคริปต์และคุณสมบัติ โดยกำหนดผ่านแท็ก <input> มีรูปแบบดังนี้

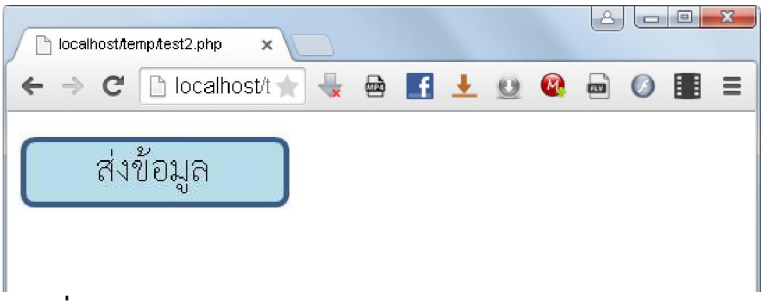
รูปแบบ

```
<input type = "image" name = "ชื่อส่วนนำเข้าข้อมูล" src = "ไดเรกทอรี/ชื่อไฟล์รูปภาพ">
```

ตัวอย่างที่ 9.6 แสดงตัวอย่างการใช้ปุ่มส่งข้อมูลแบบรูปภาพ

```
<form name= "frmPhoto" method = "POST" action = "upload.php">
    <input type ="image" name = "btnUpload" src = "parinya.jpg">
</form>
```

ผลลัพธ์



ภาพที่ 9.6 แสดงตัวอย่างการใช้งานปุ่มส่งค่าข้อมูลแบบรูปภาพ

9.2.4 กลุ่มอื่นๆ นอกจากส่วนนำเข้าข้อมูลทั้ง 3 กลุ่มข้างต้นที่ได้กล่าวไว้แล้วนั้น ในภาษา PHP ยังมีส่วนนำเข้าข้อมูลที่นิยมใช้งานบ่อยๆ อีก ได้แก่ 1) ส่วนนำเข้าข้อมูลแบบซ่อนข้อความ (Hidden Field) และ 2) ส่วนนำเข้าข้อมูลจากไฟล์ (File Field) โดยมีรายละเอียดดังนี้

1) ส่วนนำเข้าข้อมูลแบบซ่อนข้อความ นิยมใช้ในการซ่อนข้อความสำหรับส่งข้อมูลในฟอร์มไปยังไฟล์สคริปต์ที่กำหนด เพื่อนำไปประมวลผลโดยไม่ต้องการให้ผู้ใช้เห็นข้อความ ส่วนนำเข้าข้อมูลแบบซ่อนข้อความนิยมใช้กันมากในระบบการทำเว็บบอร์ดโดยซ่อนหมายเลขคำถามไว้ในฟอร์มตอบคำถาม เป็นต้น สามารถเขียนสคริปต์และคุณสมบัติ โดยกำหนดผ่านแท็ก <input> มีรูปแบบ ดังนี้

รูปแบบ

```
<input type = "hiddden" name = "ชื่อส่วนนำเข้าข้อมูล" value = "ค่าข้อมูล">
```

ตัวอย่างที่ 9.7 แสดงตัวอย่างการเขียนสคริปต์ส่วนนำเข้าข้อมูลแบบซ่อนข้อความ

```
<form name= "frmHiddenField" method = "POST" action = "test-hidden.php">
    <input type ="hidden" name = "id" value = "001">
</form>
```

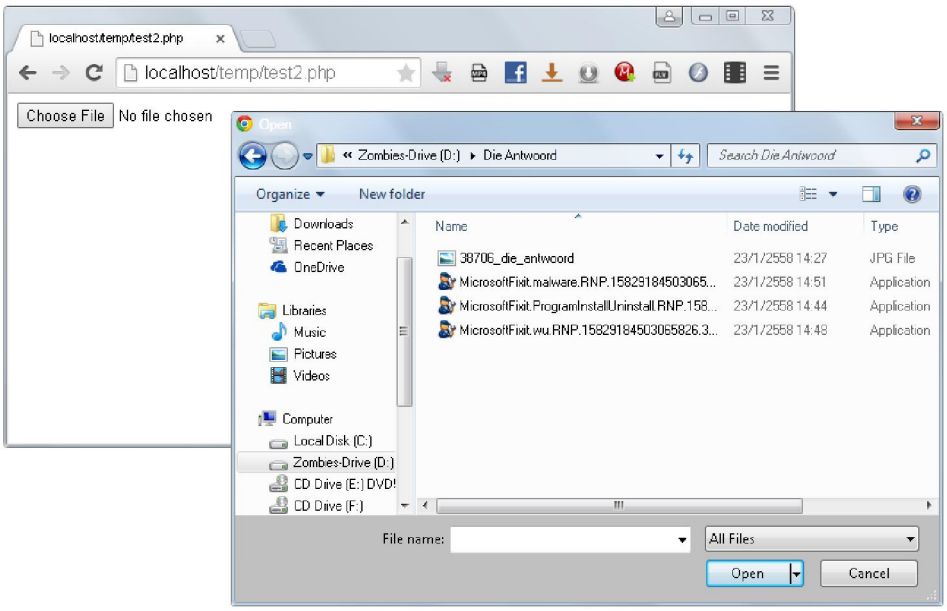
2) ส่วนนำเข้าข้อมูลจากไฟล์ ใช้สำหรับเปิดไฟล์ที่ต้องการได้ทุกนามสกุล เช่น .txt, .jpg, .gif เป็นต้น วิธีการที่สำคัญ คือ เพิ่มสคริปต์ในส่วนของ <form> คือ enctype = "multipart/form-data" เพื่อให้สามารถส่งข้อมูลรูปภาพจากส่วนนำเข้าข้อความ มีรูปแบบดังนี้

รูปแบบ

```
<input type = "file" name = "ชื่อส่วนนำเข้าข้อมูล" size = "ความกว้างของช่องรับ">
```

ตัวอย่างที่ 9.8 แสดงตัวอย่างการเขียนสคริปต์เพื่อให้สามารถรับข้อมูลจากไฟล์

```
<form name= "frmMember" method = "POST" action = "member.php" enctype = "multipart/form-data">
    <input type ="file" name = "photo">
</form>
```



ภาพที่ 9.7 แสดงตัวอย่างการใช้งานส่วนนำเข้าข้อมูลจากไฟล์

9.3 การทดสอบการแสดงผลฟอร์ม

กรณีตัวอย่างที่จะอธิบายในลำดับถัดไป จะประกอบไปด้วยไฟล์สคริปต์ PHP จำนวน 2 ไฟล์ ดังนี้ 1) regist.php ใช้สำหรับสมัครสมาชิกของระบบจากผู้ใช้งาน และ 2) member.php ใช้สำหรับแสดงผลการสมัครสมาชิก รูปแบบการเรียกใช้ดูข้อมูลจากฟอร์มจะขึ้นอยู่กับวิธีการส่งข้อมูล มีรูปแบบดังนี้

ตารางที่ 9.2 ความสัมพันธ์ระหว่างวิธีการส่งและรูปแบบการเรียกใช้ข้อมูล

วิธีส่งข้อมูล	รูปแบบการเรียกใช้ข้อมูล
POST	$_POST ["ชื่อส่วนนำเข้าข้อมูล"]
GET	$_GET ["ชื่อส่วนนำเข้าข้อมูล"]

จากตารางที่ 9.2 แสดงความสัมพันธ์ระหว่างวิธีการส่งและรูปแบบการเรียกใช้ข้อมูล จะเห็นได้ว่าวิธีการส่งข้อมูลจากฟอร์มจะมีผลต่อรูปแบบการเรียกใช้ข้อมูล ดังนั้นจะต้องตรวจสอบวิธีการส่งให้ชัดเจน ไม่เช่นนั้นอาจทำให้การเรียกใช้งานไม่ถูกต้อง นอกจากรูปแบบการเรียนใช้ข้อมูล $_POST และ $_GET แล้วยังสามารถใช้ $_REQUEST แทนวิธีการส่งข้อมูลทั้ง 2 วิธีได้

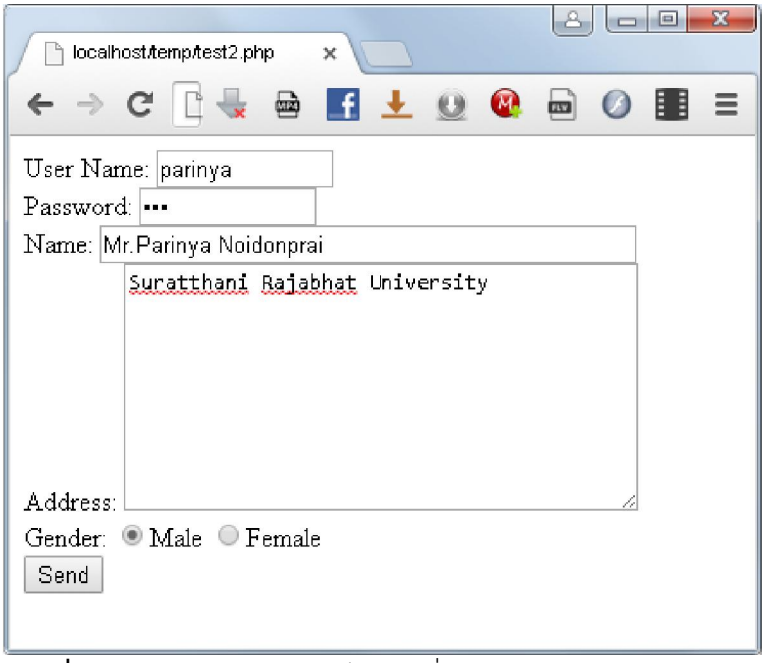
9.3.1 regist.php มีสคริปต์ดังนี้

ตัวอย่างที่ 9.9 แสดงสคริปต์ของไฟล์ regist.php

```
<form name= "frmMember" method = "POST" action = "member.php">
        User Name: <input name = "txtUser" type = "text" size = "10"><br>
        Password: <input name = "txtPass" type = "password" size = "10"><br>
        Name: <input name = "txtName" type = "text" size = "40"><br>
        Address: <textarea name = "txtAddress" cols = "40" row = "5"></textarea><br>
```

```
        Gender:
    <input type ="radio" name = "rdoGender" value = "M" checked = "checked">Male
    <input type ="radio" name = "rdoGender" value = "F">Female
    <input type ="submit" name = "btnSubmit" value = "Send">
</form>
```

ผลลัพธ์



ภาพที่ 9.8 แสดงตัวอย่างหน้าเว็บเพจเมื่อเรียกดูไฟล์ regist.php บนเว็บบราวเซอร์

จากภาพที่ 9.8 เมื่อป้อนข้อมูลต่างๆ ตามลำดับส่วนนำเข้าข้อมูลเรียบร้อยแล้ว เมื่อกดปุ่ม Send จะส่งข้อมูลทั้งหมดไปประมวลผลที่ไฟล์ member.php โดยสคริปต์ของ member.php มีรายละเอียดดังนี้

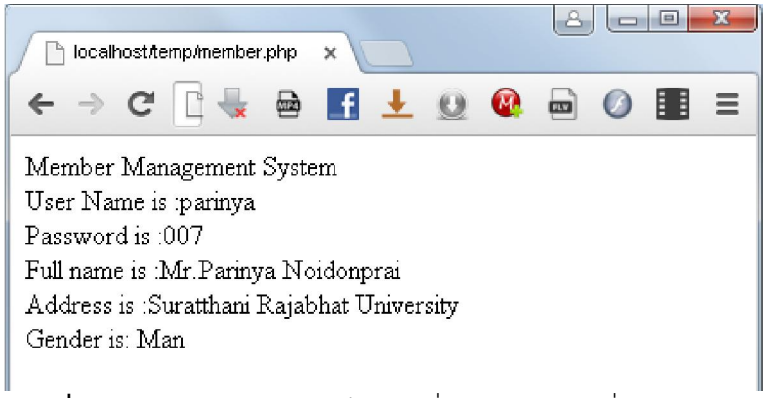
9.3.2 member.php มีสคริปต์ดังนี้

ตัวอย่างที่ 9.10 แสดงสคริปต์ของไฟล์ member.php

```
1    <?php
2        printf ("Member Management System<br>");
3        printf ("User Name is :%s <br>", $_POST["txtUser"]);
4        printf ("Password is :%s <br>", $_POST["txtPass"]);
5        printf ("Full name is :%s <br>", $_POST["txtName"]);
6        printf ("Address is :%s <br>", $_POST["txtAddress"]);
7        if ($_POST["rdoGender"]=="M") {
8            printf ("Gender is: Man");
```

```
9            } else {
10                   printf ("Gender is: Female");
11           }
13    ?>
```

ผลลัพธ์



ภาพที่ 9.9 แสดงตัวอย่างหน้าเว็บเพจเมื่อคลิก Send เพื่อส่งข้อมูลไปประมวลผลที่ member.php

จากตัวอย่างที่ 9.10 แสดงสคริปต์ของไฟล์ member.php เพื่อใช้เรียกดูข้อมูลที่ถูกส่งมาจากไฟล์ regist.php ด้วยวิธีการส่งรูปแบบ POST ดังนั้นส่วนนำเข้าข้อมูลทั้งหมดที่ส่งมาจากไฟล์ regist.php เมื่อประมวลผลในไฟล์ member.php จะถูกใส่ลงในตัวแปรพิเศษ $_POST โดยบรรทัดที่ 2-8 จะเป็นการแสดงผลค่าตัวแปรปกติ ส่วนในบรรทัดที่ 7 มีการใช้โครงสร้างเงื่อนไขเพื่อตรวจสอบค่าตัวแปร ($_POST["rdoGender"] เป็นเพราะค่าที่ถูกส่งจาก regist.php มีค่าเพียง 2 ค่า คือ M แทนค่า Man และ F แทนค่า Female ดังนั้นการแสดงผลจำนวนเป็นแปลงข้อมูลกลับเพื่อให้ผู้ใช้สามารถอ่านและแปลความหมายได้ง่าย

9.4 การจัดเก็บข้อมูลลงไฟล์

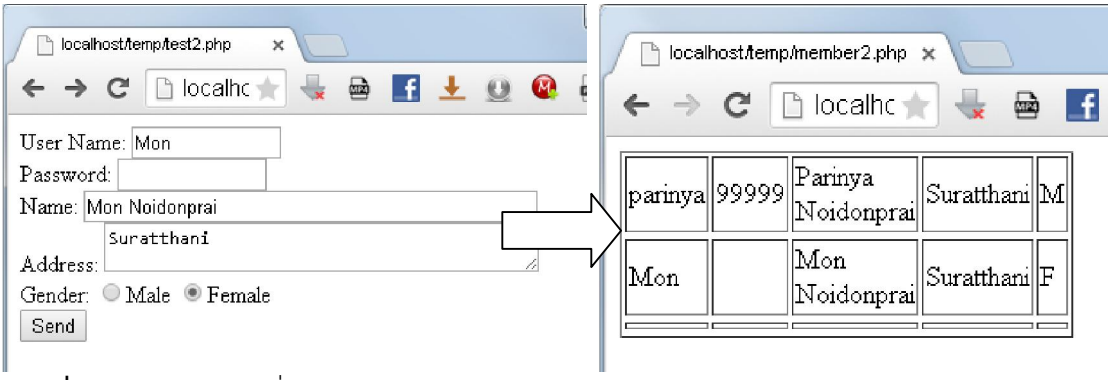
สำหรับในหัวข้อนี้จะประยุกต์เพิ่มเติมจากตัวอย่างก่อนหน้า เพื่อจัดเก็บข้อมูลสมาชิกลงในไฟล์ จะเพิ่มเติมในสคริปต์ member.php ดังนี้

ตัวอย่างที่ 9.11 แสดงสคริปต์เพื่อจัดเก็บข้อมูลลงไฟล์ เพิ่มเติมจาก member.php

```
1<?php
2     function show_member ($filemember) {
3          $fp = file ($filemember);
4          $row = explode ("-", $fp[0]);
5          echo "<table border=1 width=50%>";
6          for($var=0; $var <= count($row)-1; $var++){
7               echo "<tr>";
```

```
8                   $field = explode ("*", $row [$var]);
9                   for($col = 0; $col <= 4; $col++) {
10                          echo "<td>"; echo $field[$col]; echo "</td>";
11                  }
12                  echo "<tr>";
13          }
14          echo "</table>";
15      }
16      $filemember = "member.txt";
17      if (file_exists ($filemember)) {
18          $member = fopen ($filemember, "a+");
19          if ($member==false) die ("Can not create member.txt");
20          $table=sprintf("%s*%s*%s*%s*%s*%s-",$_POST["txtUser"],$_POST["txtPass"],
$_POST["txtName"], $_POST["txtAddress"], $_POST["rdoGender"]);
21          fwrite ($member, $table);
22          fclose ($member);
23          show_member ($filemember);
24      } else {
25          echo "<br>Creating member.txt in first time<br>";
26          $member = fopen ($filemember, "w");
27          if ($member==false) die ("Can not create member.txt");
28          $table=sprintf("%s*%s*%s*%s*%s*%s-",$_POST["txtUser"],$_POST["txtPass"],
$_POST["txtName"], $_POST["txtAddress"], $_POST["rdoGender"]);
29          fwrite ($member, $table);
30          fclose ($member);
31          show_member($filemember);
32      }
33?>
```

จากตัวอย่างที่ 9.11 แสดงสคริปต์เพื่อจัดเก็บข้อมูลลงเท็กซ์ไฟล์ โดยสามารถเพิ่มข้อมูลสมาชิกได้เรื่อยๆ พร้อมหน้ารายงานผล ดังภาพที่ 9.10



ภาพที่ 9.10 แสดงหน้าเพิ่มข้อมูลสมาชิก และหน้ารายงานผลข้อมูลสมาชิก

สรุป

การพัฒนาเว็บแอพพลิเคชันที่สมบูรณ์จะต้องมีการรับค่าข้อมูลหรือนำเข้าข้อมูลเพื่อส่งไปประมวลผลให้ได้ผลลัพธ์ตามที่ได้ออกแบบไว้ การนำเข้าข้อมูลสามารถนำเข้าผ่านทางฟอร์มของภาษา HTML โดยจะต้องระบุวิธีการส่งข้อมูลให้ถูกต้อง และกำหนดชื่อส่วนนำเข้าข้อมูลให้ชัดเจน เพราะจะมีผลต่อการเรียกใช้ในไฟล์สคริปต์ที่ระบุในฟอร์ม ซึ่งในแท็กของฟอร์มนั้นจำเป็นต้องระบุส่วนนำเข้าข้อมูลคุณสมบัติต่างๆ ที่เกี่ยวข้อง เช่น ความกว้าง จำนวนบรรทัด ลักษณะการแสดงผลตัวเลือกทั้งแบบเลือกตอบได้เพียงครั้งเดียว และแบบเลือกซ้ำได้ เป็นต้น ดังนั้นผู้พัฒนาระบบจะต้องศึกษาและพิจารณาเลือกใช้ตามความเหมาะสมของงาน

คำถามท้ายบท

1. จงอธิบายถึงรูปแบบการใช้งานแท็ก <form> และอธิบายถึงการส่งข้อมูลมีกี่วิธี มีความเหมือนและแตกต่างกันอย่างไรของแต่ละวิธี

2. จงอธิบายหลักการทำงานของสคริปต์ที่กำหนดให้ต่อไปนี้

```
<form name= "frmMember" method = "POST" action = "member.php">
        User Name: <input name = "txtUser" type = "text" size = "10"><br>
        Password: <input name = "txtPass" type = "password" size = "10"><br>
        Name: <input name = "txtName" type = "text" size = "40"><br>
        Address: <textarea name = "txtAddress" cols = "40" row = "5"></textarea>
</form>
```

3. จงอธิบายหลักการทำงานของสคริปต์ที่กำหนดให้ต่อไปนี้

```
<form name= "frmtest" method = "POST" action = "#">
    Gender:
    <input type ="radio" name = "rdoTest" value = "1">Apple
```

```
    <input type ="radio" name = "rdoTest" value = "2">Coconut
    <input type ="radio" name = "rdoTest" value = "3" checked = "checked">Pineapple
    <input type ="radio" name = "rdoTest" value = "4">Durian
</form>
```

4. จงอธิบายหลักการทำงานของสคริปต์ที่กำหนดให้ต่อไปนี้

```
<form name= "frmTest" method = "POST" action = "#">
        Provincce:
        <select name = "sltProvince">
            <option value = "1" selected = "selected">Suratthani</option>
            <option value = "2">Chumphon</option>
            <option value = "3">Ranong</option>
        </select>
</form>
```

5.จงอธิบายและยกตัวอย่าง ส่วนนำเข้าข้อมูลกลุ่มปุ่มคำสั่ง