

برقراری ارتباط با کاربر

واحد کار هفتم

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

- ۱ - اصول دریافت اطلاعات از فرم‌های HTML را بداند.
- ۲ - اصول استفاده از متدهای GET و POST را در فرم بشناسد.
- ۳ - بتواند داده‌های ورودی را اعتبارسنجی کند.
- ۴ - بتواند از طریق PHP ایمیل ارسال کند.
- ۵ - بتواند یک فایل را از طریق فرم، Upload کند.

زمان(ساعت)	
عملی	نظری
۸	۴

مقدمه

با مفهوم فرم در مباحث طراحی صفحات وب مقدماتی آشنا شدید، فرم‌ها اساساً برای ارسال اطلاعات از سرویس گیرنده به سرویس دهنده طراحی شده‌اند و به دلیل این که PHP یک زبان سمت سرویس دهنده است، باید بتوانید فرم‌ها را در PHP مدیریت کنید.

۱-۷ دریافت اطلاعات از فرم

همان‌طور که می‌دانید فرم‌ها برای ارسال اطلاعات از دو متد GET و POST استفاده می‌کنند.

سؤال: تفاوت متدهای GET و POST چیست؟

کد زیر یک فرم ایجاد می‌کند:

```
<html>
<head>
<title>form</title>
</head>
<body>

<form method="POST" action="Page1.php" name="Form1">
<input type="text" name="txt1" size="20">
<input type="submit" value="send" name="B1">
<input type="reset" value="cancel" name="B2">
</form>

</body>
</html>
```

کد فوق را در یک فایل نوشته و آن را با نام index و با پسوند htm در مسیر سرویس دهنده وب محلی ذخیره کنید. در این کد ۳ شیء ورودی وجود دارد، یک شیء کادر متنی و یک دکمه از نوع Submit برای ارسال اطلاعات به صفحه‌ای که در خاصیت action از فرم تعیین شده است (Page1.php) و یک دکمه از نوع reset برای پاک کردن محتویات فرم.

حال می‌خواهیم در صفحه Page1.php متغیر ارسال شده از سوی فایل قبل را دریافت کرده و آن را ۳ بار در خروجی چاپ کنیم.
کد فایل Page1.php به صورت زیر نوشته می‌شود:

```
<?php  
$str1 = $_POST['txt1'];  
for($i = 0 ; $i < 3 ; $i++)  
{  
echo ($str1."<br>");  
}  
?>
```



شکل ۷-۱

حال فایل index.htm را اجرا کنید.
یک کلمه یا عبارت را به دلخواه درون کادر متنی شکل ۷-۲ بنویسید و روی دکمه Send کلیک کنید.



شکل ۷-۲



شکل ۷-۳

به نوار Address شکل ۳-۷ توجه کنید، صفحه Page1.php در حال نمایش است، در این صفحه کلمه وارد شده در کادر متنی شکل ۲-۷، سه بار نوشته شده است. به کد فایل Page1.php دقت کنید:

متغیری که در کادر متنی فرم با عنوان txt1 نامگذاری شده بود توسط متد POST برای Page1.php ارسال شده است.

این متغیر ارسال شده در Page1.php با عنوان \$_POST['txt1'] قابل شناسایی است و چون یک متغیر است باید ابتدای آن علامت \$ درج شود.

برای دسترسی راحت تر به این مقدار، بهتر است در یک متغیر مشخص قرار داده شود.

```
$str1 = $_POST ['txt1'];
```

حال در Page1.php می توان با آن مانند یک متغیر عادی برخورد کرد که در اینجا توسط حلقه for آن را ۳ بار چاپ کرده ایم.

نکته اگر متغیرهای فایل مبدأ توسط متد GET ارسال شوند، در صفحه مقصد باید به صورت ['نام فیلد'] GET شناسایی شوند.

۱-۱-۷ تابع REQUEST_\$

PHP یک تابع درون ساخته مهم دارد که می تواند صرف نظر از روش استفاده شده برای ارسال اطلاعات توسط فرم به مقادیر \$_GET و \$_POST دسترسی داشته باشد. این تابع می تواند در متدهای ارسال get و post استفاده شود. اگر مثال قبل با استفاده از تابع \$_REQUEST نوشته شود، کد فایل به صورت زیر خواهد بود:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

</head>

<body>

Welcome <?php echo \$_REQUEST["txt1"]; ?>!

</body>

</html>

۲-۷ اعتبارسنجی داده‌های ورودی


تعیین صحت و درستی داده‌های ورودی از طرف کاربران یکی از مهم‌ترین جنبه‌های امنیت سایت است. داده‌ای که به‌طور اشتباه از طرف کاربر وارد می‌شود، می‌تواند بر اثر اشتباه تایپی باشد، یا این‌که کاربر یک هکر بوده و قصد نفوذ یا آسیب رساندن به سایت را داشته باشد، لذا داده حتماً باید پس از نوشتن توسط کاربر اعتبارسنجی شود.

فرض کنید در یک فرم طراحی شده از کاربر درخواست شده است اطلاعات مربوط به خودش را وارد کند، اگر این اطلاعات مانند نام و نام‌خانوادگی باشد، فقط هر ترکیبی از حروف الفبا که بیش از یک حرف داشته باشد قابل قبول است اما اگر کاربر یک عدد یا تنها یک حرف الفبا وارد کند، نشان‌دهنده اشتباه بودن اطلاعات ورودی کاربر است یا به عنوان مثال اگر از کاربر بخواهیم رشته تحصیلی خود را وارد کند و کاربر abcd را وارد کند، نشان‌دهنده اشتباه بودن اطلاعات ورودی کاربر است. در این‌گونه مواقع می‌توان با در نظر گرفتن حالات مختلفی هنگام طراحی تا حدود زیادی از وارد کردن اطلاعات نادرست از طرف کاربر جلوگیری کرد. مثلاً برای ورودی‌هایی مانند نام و نام‌خانوادگی بررسی کرد که عبارت ورودی از طرف کاربر ترکیبی از چند حرف الفبا باشد، برای شماره تلفن ترکیبی از اعداد پذیرفته شود و برای ورودی‌هایی مانند رشته تحصیلی یا محل زندگی به دلیل محدود و مشخص بودن رشته‌های تحصیلی و اسامی شهرها، از لیست‌های انتخاب استفاده کرد که کاربر یکی از موارد موجود در لیست را که با اطلاعات او تطابق دارد، انتخاب کند.

به این ترتیب می‌توان تا حد زیادی مطمئن بود اطلاعات ورودی در پایگاه داده خطرناک نخواهند بود. یکی از راه‌هایی که هکرها و افرادی که قصد تخریب اطلاعات سایت را دارند، انتخاب می‌کنند این است که سعی کنند به اطلاعات ارسال شده فرم از طریق متد POST دسترسی پیدا کرده و آن‌ها را تغییر دهند. پس از تغییر اطلاعات توسط هکرها اغلب شرایطی پیش می‌آید که داده‌ها به شکل نامعتبری درمی‌آیند، لذا با بررسی اعتبار داده‌ها می‌توان از راه‌یابی داده‌های مشکوک به بانک اطلاعاتی جلوگیری کرد.

۳-۷ ارسال نامه الکترونیکی از طریق PHP

یکی از قابلیت‌های PHP ارسال نامه الکترونیکی است. برای این‌که این قابلیت در اختیار برنامه‌نویس قرار بگیرد باید پیکربندی فایل php.ini انجام گیرد.

نکته  در صورتی که از سرویس‌دهنده WAMP استفاده می‌کنید، نیاز به پیکربندی نیست و خود WAMP به‌طور خودکار تمام سرویس‌ها را آماده می‌کند.

برای پیکربندی فایل php.ini، روی آیکن نیم‌دایره سرویس‌دهنده WAMP کلیک کنید، ماوس را روی گزینه config files نگه داشته و از زیر منوی باز شده روی گزینه php.ini کلیک کنید.

با کلیک روی گزینه Edit از نوار منو و انتخاب گزینه Find، کادر محاوره Find ظاهر می‌شود. در کادر متنی Find what عبارت mail function را وارد کرده و با کلیک روی دکمه Find Next، این عبارت را در کل فایل جستجو کنید.

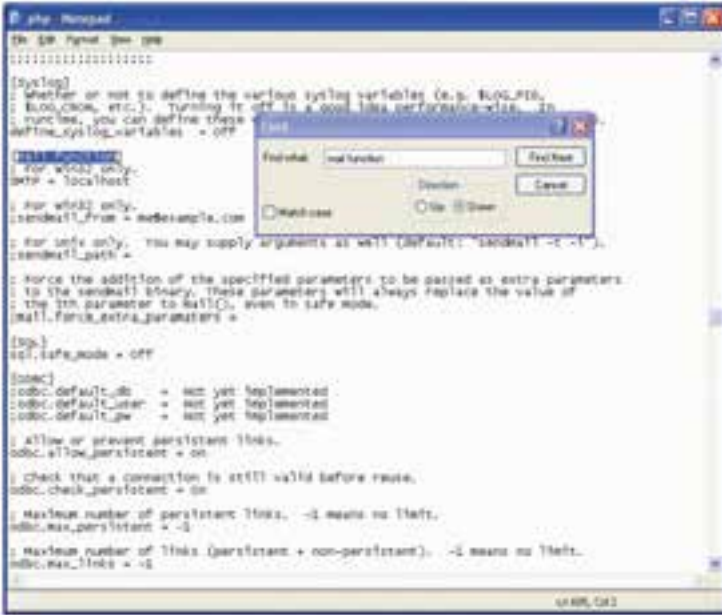
پس از عبارت [mail function] سه گزینه با عناوین SMTP، sendmail_from و sendmail_path وجود دارد.

برای پیکربندی در ویندوز مقابل گزینه SMTP باید آدرس سرویس‌دهنده پست الکترونیکی، برای sendmail_path مسیر برنامه ارسال پست الکترونیکی و هم‌چنین برای sendmail_from آدرس پست الکترونیکی خود را مشخص نمایید.

[mail function]

SMTP = SMTP.my.isp.net

sendmail_from = myaccount@yahoo.com



شکل ۷-۴

۱-۳-۷ به کارگیری توابع ارسال ایمیل

پس از اطمینان از پیکربندی فایل `php.ini`، برای ارسال ایمیل می‌توانید از دستور `mail` استفاده کنید.

ساختار کلی این تابع به صورت زیر است:

`bool mail (string to , string object , string message);`

در صورتی که ایمیل به درستی و بدون اشکال ارسال شود، این تابع مقدار `True` را برمی‌گرداند و در غیر این صورت مقدار برگشتی تابع، `False` خواهد بود.

پارامترهای تابع `mail`

- نکته** • برنامه ارسال نامه الکترونیکی روی سیستم به صورت محلی اجرا نمی‌شود و این کدها حتماً باید روی سرویس‌دهنده وب واقعی اجرا شوند.
- در اغلب سرویس‌دهنده‌های PHP امکان ارسال نامه الکترونیکی به طور رایگان امکان‌پذیر است اما برای اطمینان بیشتر قبل از اینکه این تابع را در برنامه استفاده کنید، از نصب `mail server` روی سرویس‌دهنده خود مطمئن شوید.

پارامترهای این تابع در جدول زیر توضیح داده شده‌اند.

عملکرد	پارامتر
توسط این پارامتر آدرس پست الکترونیکی دریافت کننده نامه تعیین می‌شود. این پارامتر می‌تواند آدرس پست الکترونیکی چند نفر را دریافت کند. اگر بخواهید یک ایمیل را برای چند نفر ارسال کنید، باید آدرس‌های ایمیل با کاراکتر ، از یکدیگر جدا شوند.	to
مثال: "user1@yahoo.com" مثال: "user1@yahoo.com , user2@yahoo.com"	
این پارامتر عنوان نامه را تعیین می‌کند.	subject
متن اصلی نامه را معین می‌کند و می‌تواند حداکثر تا ۷۰ کاراکتر را دریافت کند.	message
این پارامتر نیز اختیاری است و چنانچه بخواهید غیر از پارامترهای فوق، پارامتر دیگری به نامه الکترونیکی اضافه کنید، می‌توانید در این بخش درج نمایید.	parameters

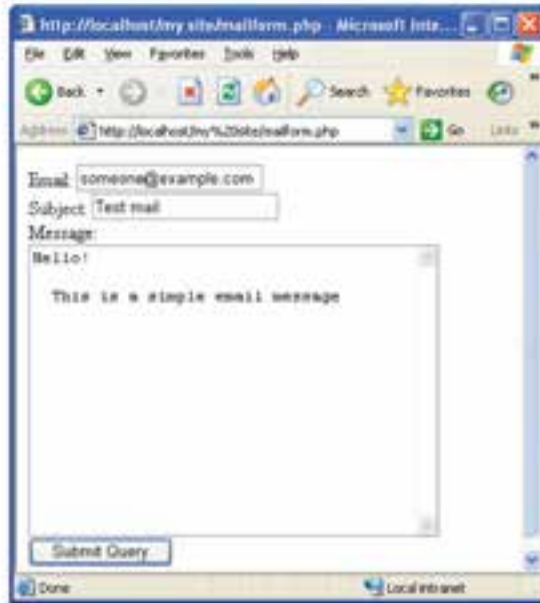
مثال:

مثال زیر یک نامه الکترونیکی ساده را در صفحه وب ارسال می‌کند:

```
<?php
$to = "someone@example.com";
$subject = "Test mail";
$message = "Hello! This is a simple email message.";
$from = "someone@example.com";

mail($to,$subject,$message);
echo "Mail Sent.";
?>
```

در یک صفحه وب بهتر است فرم مخصوصی مانند شکل ۵-۷ به منظور دریافت پارامترهای نامه الکترونیکی و سپس ارسال آن طراحی شود.



شکل ۷-۵ نمونه فرم ارسال نامه الکترونیکی

نکته چنانچه در صورت نصب بودن mail server توسط برنامه، نامه الکترونیکی ارسال نشود، در فایل php.ini از ابتدای عبارت sendmail_path علامت ; را بردارید و مسیر mail server را مقابل آن نوشته و هم‌چنین مقدار smtp_port را برابر با ۲۵ قرار دهید و تغییرات را ذخیره کنید.

۷-۴ بارگذاری (Upload) فایل‌ها از طریق فرم

کاربر می‌تواند توسط تابع ftp_put یک فایل موجود روی سیستم خود را توسط فرم upload کند.

شکل کلی این دستور به صورت زیر است:

```
bool ftp_put (resource $ftp_stream , string $remote_file , string $Local_file , int $mode);
```

پارامترهای تابع ftp_put

ftp_stream: یک اتصال FTP برقرار می‌کند.

remote_file: مسیر و نام فایل راه دور را تعیین می‌کند.

Local_file: مسیر و نام فایل محلی را تعیین می‌کند.
mode: شیوه انتقال را تعیین می‌کند که می‌تواند یکی از مقادیر FTP_ASCII یا FTP_BINARY را بپذیرد.
اگر بارگذاری فایل با موفقیت انجام شود، تابع مقدار True را برمی‌گرداند، در غیر این صورت مقدار برگشت داده شده توسط این تابع False خواهد بود.

مطالعه آزاد

ایجاد فرم بارگذاری فایل

برای بارگذاری یک فایل در اینترنت نیز باید آن را از کاربر دریافت نمود، دریافت فایل از کاربر و ارسال آن از طریق فرم انجام می‌شود.
به منظور انتخاب فایل برای بارگذاری، باید نوع عنصر input را از نوع فایل قرار دهید:

```
<input type="file" name="file" id="file" />
```

در این صورت دکمه Browse نیز روی فرم ظاهر می‌شود که امکان انتخاب فایل از طریق کادر محاوره Choose file را فراهم می‌کند.
هنگام طراحی فرم بارگذاری فایل، باید enctype را از نوع multipart/form-data قرار دهید، این نوع کد گذاری برای ارسال اطلاعات باینری مانند محتویات فایل مناسب است.

مثال: کد زیر یک فرم انتخاب فایل برای بارگذاری را ایجاد می‌کند:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>  
</head>  
<body>  
  
<form action="upload_file.php" method="post"
```

```
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file" />
<br />
<input type="submit" name="submit" value="Submit" />
</form>

</body>
</html>
```

با اجرای کد فوق، مرورگر مشابه شکل ۲-۵ یک فایل را از کاربر دریافت می‌کند.



شکل ۶-۷

در این کد، فایل دریافت شده، برای اسکریپت `upload_file.php` ارسال می‌شود. به منظور پیاده‌سازی اسکریپت `upload_file.php` می‌توانید کد بعد را تایپ کنید و آن را با نام `upload_file.php` ذخیره نمایید.

```
<?php
if ($_FILES["file"]["error"] > 0)
{
echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
```

```
echo "Upload: " . $_FILES["file"]["name"] . "<br />";  
echo "Type: " . $_FILES["file"]["type"] . "<br />";  
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";  
echo "Stored in: " . $_FILES["file"]["tmp_name"];  
}  
?>
```

اگر به طور مثال اسم فایل بارگذاری شده test1.htm باشد، نتیجه اجرای اسکریپت upload_file.php به صورت زیر خواهد بود:

Upload: test1 htm

Type: text/html

Size: 0.14609375 Kb

Stored in: C:\wamp\tmp\php19.tmp

فرم بارگذاری فایل، چند آرگومان را به اسکریپت PHP ارسال می کند. برای شناسایی هر کدام از این آرگومان ها می توان از تابع مشخصی استفاده نمود. توابع مهم و کاربردی فرم بارگذاری عبارتند از:

• **\$_FILES["file"]["error"]**: نتیجه خطای احتمالی، هنگام بارگذاری فایل را مشخص می کند.

اگر هنگام بارگذاری فایل هیچ خطایی رخ نداده باشد، این تابع مقدار صفر را برمی گرداند.

• **\$_FILES["file"]["name"]**: نام فایل بارگذاری شده را مشخص می کند.

• **\$_FILES["file"]["type"]**: نوع و پسوند فایل بارگذاری شده را مشخص می کند. به

عنوان مثال برای فایل های تصویری می تواند image/ jpg، image/tiff، image/ gif و ... باشد.

• **\$_FILES["file"]["size"]**: اندازه فایل بارگذاری شده را برحسب بایت مشخص

می کند.

• **\$_FILES["file"]["tmp_name"]**: مسیر ریشه فایل های ذخیره شده مربوط به

سرویس دهنده را مشخص می کند. تا زمانی که فایل ها توسط سرویس دهنده WAMP اجرا

می شوند، این فایل ها در مسیر نصب سرویس دهنده WAMP ذخیره می شوند.



نکته فراهم کردن امکان بارگذاری فایل توسط کاربر در وب سایت به دلیل احتمال وجود تهدیدات امنیتی می تواند خطرات جدی در پی داشته باشد، بنابراین فقط باید در صورت ضرورت از این قابلیت در صفحات وب استفاده کنید.

مثال قبل شکل بسیار ساده ای از بارگذاری فایل است، شما به عنوان طراح وب سایت می توانید محدودیت هایی را برای فایل انتخاب شده از طرف کاربر تعیین کنید. به طور مثال اگر در صفحه وب از کاربر خواسته باشید عکس خودش را بارگذاری کند، می توانید از او بخواهید تصویر ارسالی خود را از نوع gif، bmp یا jpg با اندازه محدود انتخاب کند، سپس در کد اسکریپت upload_file.php نوع و اندازه آن را بررسی کنید و در صورت تناقض به فایل اجازه بارگذاری روی سرور ندهید. در این صورت می توانید مشابه کد زیر را برای اسکریپت upload_file.php بنویسید:

```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpg")
|| ($_FILES["file"]["type"] == "image/bmp")))
&& ($_FILES["file"]["size"] < 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
}
```

```
}  
else  
{  
    echo "Invalid file";  
}  
?>
```

تمرین

از کاربر بخواهید در یک صفحه وب رزومه کاری خود را در قالب فایل متنی با پسوند .doc یا .rtf و اندازه حداکثر ۱۰۰ کیلوبایت، ارسال کند و اگر فایل انتخاب شده کاربر این ویژگی‌ها را نداشت، پیغام مناسبی نمایش داده شود.

ذخیره فایل بارگذاری شده

فایل‌های کپی شده در مسیر ریشه، پس از خاتمه اسکریپت، حذف می‌شوند، به همین دلیل اگر بخواهید فایل بارگذاری شده از طرف سرویس‌گیرنده را در سرویس‌دهنده استفاده کنید، باید عمل انتقال فایل و ذخیره آن را در مسیری غیر از مسیر ریشه سرویس‌دهنده نیز انجام دهید.

برای انتقال فایل‌ها از ریشه، تابع `move_uploaded_file` مورد استفاده قرار می‌گیرد. برای این کار می‌توانید کد زیر را به انتهای اسکریپت `upload_file.php` اضافه کنید:

```
move_uploaded_file($_FILES["file"]["tmp_name"],  
"upload/" . $_FILES["file"]["name"]);  
echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
```

هنگام کپی فایل روی سرویس‌گیرنده ممکن است فایل هم نام و هم نوع با فایل کاربر وجود داشته باشد، در نتیجه کپی فایل کاربر باعث از بین رفتن فایل هم نامی که از قبل وجود داشته، می‌شود، برای پیشگیری از این مشکل می‌توانید از تابع `file_exists` برای بررسی تکراری نبودن فایل استفاده کنید.

توصیه: قبل از انجام عمل کپی و نوشتن اسکریپت مربوط به آن در مسیر ذخیره وب سایت خود، پوشه‌ای را برای ذخیره فایل‌های بارگذاری شده ایجاد کنید.

مثال:

کد زیر در صورت عدم وجود خطا، مشخصات فایل ارسال شده توسط کاربر را در خروجی نشان می‌دهد، سپس اگر فایل تکراری بود، پیغام already exists به همراه نام فایل ظاهر می‌شود، در غیر این صورت عمل انتقال آن به پوشه upload که آن را در مسیر سرویس‌دهنده ایجاد کرده‌ایم، انجام می‌شود.

```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpg")
|| ($_FILES["file"]["type"] == "image/bmp")))
&& ($_FILES["file"]["size"] < 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

if (file_exists("upload/" . $_FILES["file"]["name"]))
{
echo $_FILES["file"]["name"] . " already exists. ";
}
}
```

```
else
{
    move_uploaded_file($_FILES["file"]["tmp_name"],
"upload/" . $_FILES["file"]["name"]);
    echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
}
}
}
else
{
    echo "Invalid file";
}
?>
```

شکل زیر نتیجه اجرای کد فوق را هنگام بارگذاری فایل تصویری sun.gif نشان می‌دهد.



شکل ۷-۷

۵-۷ طراحی فرم بازخورد (Feedback)

یکی از قابلیت‌هایی که در بسیاری از سایت‌ها استفاده می‌شود، ارائه فرم بازخورد است، این گونه فرم‌ها بیشتر به منظور دریافت نظرات کاربران در مورد یک موضوع خاص به کار می‌رود.

به عنوان مثال می‌توان به سایت‌های صدا و سیما اشاره کرد که پس از پخش یک فیلم سینمایی یا یک سریال تلویزیونی فرم‌هایی را روی سایت خود به منظور بررسی تأثیرات آن بر مخاطب، قرار می‌دهند.

این فرم‌ها حاوی سؤالاتی در مورد فیلم یا سریال هستند که کاربران با پر کردن آن‌ها و ارسال فرم پر شده به سرویس دهنده نظرات خود را در اختیار مدیر سایت قرار می‌دهند.

در طراحی یک فرم بازخورد نکات متعددی را باید در نظر گرفت:

- از طراحی سؤالات با تعداد زیاد خودداری کنید.
 - سؤالات طرح شده دارای جواب‌های کوتاه باشند.
 - کاربر بتواند پس از ارسال نظر خود، آمارسنجی نظرات سایر کاربران را ببیند.
 - کاربر بتواند انعکاس نظرات خود را به‌طور عملی ببیند.
- شکل‌های ۷-۸ و ۷-۹ دو نمونه فرم بازخورد را به شما نشان می‌دهند.



شکل ۷-۸



شکل ۷-۹

واژه‌نامه

Feedback

بازخورد

خلاصه مطالب

- برای دریافت اطلاعات از سروریس گیرنده و ارسال آن‌ها به سروریس دهنده از فرم‌ها استفاده می‌شود.
- ارسال اطلاعات در فرم به دو شیوه GET و POST صورت می‌گیرد.
- برای شناسایی متغیرهای ارسالی از طرف فایل مبدأ در فایل مقصد از توابع GET و POST استفاده می‌شود.
- اعتبارسنجی داده‌های ارسال شده یکی از مهم‌ترین جنبه‌های امنیت سایت است، زیرا هکرها با دسترسی به اطلاعات ارسالی و تخریب آن‌ها غالباً کدهای نامعتبری تولید می‌کنند که با بررسی صحت و اعتبار آن‌ها تا حدود زیادی می‌توان از حمله افراد هکر و ... جلوگیری کرد.
- برای ارسال نامه الکترونیکی از طریق ایمیل باید تنظیماتی روی فایل php.ini انجام شود، اما اگر از WAMP استفاده می‌کنید، این تنظیمات به‌طور خودکار انجام می‌شوند.
- ارسال ایمیل از طریق تابع Mail صورت می‌گیرد.

آزمون نظری

- ۱ - اگر در ارسال اطلاعات از فایل a.php به فایل b.php از متد POST استفاده شده باشد متغیرهای دریافتی توسط چه عبارتی شناخته می‌شوند؟
- الف - POST
ب - _POST
ج - GET
د - _GET
- ۲ - در پیکربندی فایل php.ini برای ارسال ایمیل، آدرس سرویس‌دهنده پست الکترونیکی مقابل کدام گزینه نوشته می‌شود؟
- الف - sendmail_path
ب - sendmail_from
ج - SMTP
د - function
- ۳ - تابع Mail در صورتی که با موفقیت نامه را ارسال کند چه مقداری را به سرویس‌دهنده می‌فرستد؟
- الف - True
ب - False
ج - متن نامه الکترونیکی
د - موضوع نامه الکترونیکی
- ۴ - در تابع بارگذاری فایل‌ها از طریق فرم، پارامتر mode چه مقادیری را می‌تواند بپذیرد؟
- الف - FTP_ASCII
ب - FTP_BINARY
ج - True
د - الف و ب

آزمون عملی

- ۱ - در یک صفحه وب نام، نام خانوادگی، شماره شناسنامه و ایمیل کاربر را دریافت کنید و اطلاعات دریافتی را در فایل دیگر توسط مرورگر نمایش دهید.
- ۲ - تابعی بنویسید که اطلاعات دریافت شده از کاربر در سؤال اول را به ایمیل او ارسال کند.
- ۳ - تابع سؤال دوم را طوری تغییر دهید که اطلاعات برای چند نفر به طور گروهی ارسال شود (آدرس‌های ایمیل را به طور اختیاری وارد کنید).