طراحی برنامه های تحت وب کار با فرم ها

فصل پنجم

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

۱- عملکرد توابع $_POST، $_GET و $_REQUEST را شرح دهد.

۲- از توابع $_POST، $_GET و $_REQUEST در برنامه استفاده کند.

۳- یک فرم برای بارگذاری فایل ایجاد کند.

۴- فایل‌ها را پس از بارگذاری در مسیر ریشه، درمسیر دیگری ذخیره کند.

در جلـد اول این کتاب با فرم‌ها و بخش‌های آن آشـنا شـدید، همان‌طور کـه می‌دانید، یکی از مهم‌ترین مسائلی که هنگام طراحی سایت پویا لازم است در نظر گرفته شود این است که اطلاعات باید به وسـیله فرم‌ها ازکاربر دریافت شوند و سـپس به منظور تحلیل و پاسخگویی مناسب برای اسکریپت PHP ارسال شوند.

معمولاً ساختار فرم و اجزاء آن در فایل HTML ایجاد می‌شود.

۱-۵ تابع $_POST$

نمونه کد زیر در فایل HTML یک فرم با دو جعبه متن و یک دکمه submit برای ارسال اطلاعات 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="welcome.php" method="post">
Name: <input type="text" name="fname" />
<p>
Age: <input type="text" name="age" />
</p>
<p>
<input type="submit" />
</p>
</form>

</body>
</html>
```

این فایل نام و سـن کاربر را دریافت، و توسـط متد post آن را برای فایل welcome.php ارسال می‌کند.

مثال:

فرض کنید کاربر اطلاعات فرم را مانند شکل زیر پر کرده باشد.

شکل ۱-۵

در فایل welcome.php با استفاده از تابع درون ساخته $_POSTآرگومان‌های ارسال شده توسط کاربر، قابل دسترسی هستند.

```
Welcome <?php echo $_POST["fname"]; ?>!
<br />
You are <?php echo $_POST["age"]; ?> years old.
```

در این صورت خروجی مشابه عبارت زیر خواهد بود.

خروجی:

```
Welcome hadi!
You are 16 years old.
```

۵-۲ تابع $_GET $

اگر متدی که فرم از آن برای ارسال اطلاعات استفاده می‌کند، get باشد نیز تابع $_GET می‌تواند عملکردی مشابه تابع $_POST در مثال فوق داشته باشد و برای دسترسی به مقادیر ارسال شده توسط فرم به کار برده شود.

فرض کنید فرم مثال فوق از متد get برای ارسال اطلاعات استفاده کرده باشد، در این صورت فایل welcome.php به صورت زیر به عناصر فرم دسترسی خواهد داشت:

```
Welcome <?php echo $_GET["fname"]; ?> !
<br />
You are <?php echo $_GET["age"]; ?> years old!
```

۵-۳ تابع $_ REQUEST

PHP یک تابع درون ساخته مهم دارد که می‌تواند صرفنظر از متد استفاده شده برای ارسال اطلاعات توسط فرم به مقادیر $_GET و $_POST و همچنین مقدار $_ COOKIE دسترسی داشته باشد.

این تابع می‌تواند در متدهای ارسال get و post استفاده شود.

اگر فایل welcome.php را در مثال قبل با استفاده از تابع $_REQUEST نوشته شود، کد فایل عبارت است از:

```
<! 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["fname"]; ?>!
<br />
You are <?php echo $_REQUEST["age"]; ?> years old.
</body></html>
```

۵-۴ ایجاد فرم بارگذاری فایل

برای بارگذاری یک فایل در اینترنت نیز بایدآن 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 ایــن صورت دکمه Browse نیز روی فرم ظاهر می‌شــود که امکان انتخاب فایل از طریق کادر محاوره file Choose را فراهم می‌کند.

هنگام طراحی فرم بارگذاری فایل، باید enctype را از نوع data-form/multipart قرار دهید، این نوع کد گذاری برای ارسال اطلاعات باینری مانند محتویات فایل مناسب است.

مثال:

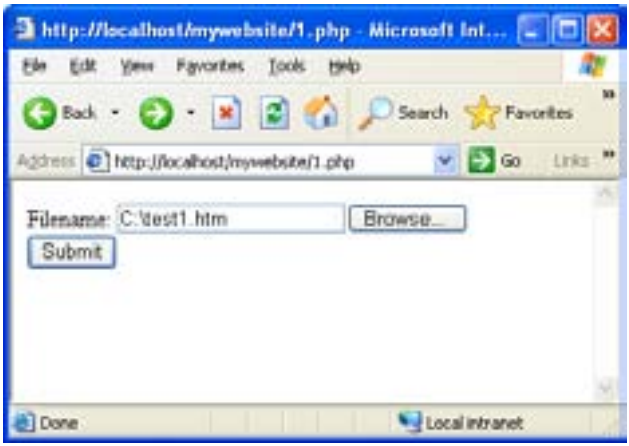
کد زیر یک فرم انتخاب فایل برای بارگذاری را ایجاد می‌کند:

```
<! 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 ۲-۵

در این کد، فایل دریافت شده، برای اسکریپت file_ upload.php ارسال می‌شود.

به منظور پیاده‌سازی اسکریپت file_ upload.php می‌توانید کد زیر را تایپ کنید و آن را با نام file_ upload.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 file_ upload.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 می‌کند. برای شناسایی هر کدام از این آرگومان‌ها می‌توان از تابع مشخصی استفاده نمود.

توابع مهم و کاربردی فرم بارگذای عبارتند از:

- **$_FILES["file"]["error"]**:نتیجه خطای احتمالی، هنگام بارگذاری فایل را مشخص می کند. اگر هنگام بارگذاری فایل هیچ خطایی رخ نداده باشد، این تابع مقدار صفر را بر می گرداند.
- **$_FILES ["file "]["name"]**: نام فایل بارگذاری شده را مشخص می‌کند.
- **$_FILES ["file"]["type "]**: نوع و پسوند فایل بارگذاری شده را مشخص می‌کند. به عنوان مثال برای فایل‌های تصویری می تواند image/jpg،image/tiff،image/gif و ... باشد.
- **$_FILES["file"]["size"]**:اندازه فایل بارگذاری شده را بر حسب بایت مشخص می کند.
- **$_FILES ["file"]["tmp_ name"]** : مسیر موقتی فایل های ذخیره شده مربوط به سرویس دهنده را مشخص می‌کند. تا زمانی که فایل‌ها توسط سرویس دهنده WAMP اجرا می‌شوند، این فایل‌ها در مسیر نصب سرویس دهنده WAMP ذخیره می‌شوند.

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

مثال فوق شکل بسیار ساده‌ی از بارگذاری فایل است، شما به عنوان طراح وب سایت می‌توانید، محدودیت‌هایی 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، می‌توانید از او بخواهید تصویر ارسالی‌اش را از نوع .gif، .bmp و یا .jpg با اندازه محدود را انتخاب کند، سپس در کد اسکریپت file_ upload.php نوع و اندازه آن را بررسی کنید و در صورت تناقض به فایل اجازه بارگذاری بر روی سرویس دهنده را ندهید.

در این صورت می‌توانید مشابه کد زیر را برای اسکریپت file_ upload.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: در یک صفحه وب از کاربر بخواهید رزومه کاری‌اش 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.

۵–۵ ذخیره فایل بارگذاری شده

مسیر ریشه یک مسیر موقتی برای مرورگر است و فایل‌های کپی شده در این مسیر، پس از خاتمه اجرای فایل و بسته شدن مرورگر حذف می‌شوند، به همین دلیل اگر فایل بارگذاری شده از طرف سرویس گیرنده را بخواهید در سرویس دهنده استفاده کنید، باید عمل انتقال فایل و ذخیره آن در مسیری غیر از مسیر ریشه سرویس دهنده را 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 file_ upload.php اضافه کنید:

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

هنگام کپی فایل برروی سرویس گیرنده ممکن است فایلی 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 به منظور بررسی تکراری نبودن فایل استفاده کنید.

- توصیه: بهتر است 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 داده می‌دهد، سپس اگر فایل تکراری بود، پیغام already exists به همراه نام فایل ظاهر می‌شود، درغیر این‌صورت عمل انتقال آن به پوشه upload که آن را در مسیر سرویس دهنده 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 را نشان می دهد.

شکل ۳-۵

مطالعه آزاد

۶-۵ نامه‌های الکترونیکی

در یک وب سایت برای ارتباط دو طرفه و تعاملی با کاربر بهتر است امکان ارسال نامه الکترونیکی به طور مستقیم فراهم شود.

در زبان اسکریپت نویسی PHP این امکان توسط تابع mail () فراهم شده است. شکل کلی این تابع به صورت زیر است:

```
mail(to,subject,message,headers,parameters);
```

پارامترهای این تابع در جدول زیر توضیح داده شده‌اند.

پارامتر	عملکرد
to	**توسط این پارامتر آدرس پست الکترونیکی دریافت کننده نامه تعیین می شود. این پارامتر می تواند آدرس پست الکترونیکی چند نفر را دریافت کند.**
subject	**این پارامتر عنوان نامه را تعیین می کند.**
message	**متن اصلی نامه را معین می کند و می تواند حداکثر تا ۷۰ کاراکتر را دریافت کند**
headers	**این پارامتر اختیاری است و بخش های آدرس پست الکترونیکی فرستنده نامه، Cc و Bcc را به نامه الکترونیکی اضافه می کند.**
parameters	**این پارامتر نیز اختیاری است و چنانچه بخواهید غیر از پارامترهای فوق، پارامتر دیگری به نامه الکترونیکی اضافه کنید، می توانید در این بخش درج نمایید.**


نکته: برنامه ارسال نامه الکترونیکی بر روی سیستم به صورت محلی اجرا نمی‌شود و این کدها یا باید روی سرویس دهنده وب واقعی اجرا شوند.و یا اینکه یکی از نرم‌افزارهای سرویس دهنده پست الکترونیکی مانند MDaemon mail server ,kerio mail server و ... را روی سیستم فرد نصب و راه‌اندازی نمایید.


نکته: در اغلب سرویس دهنده‌های PHP امکان ارسال نامه الکترونیکی به طور رایگان امکان پذیر است اما برای اطمینان بیشتر قبل از اینکه این تابع را در برنامه استفاده کنید، از نصب mail server روی سرویس دهنده خود مطمئن شوید.

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


مثال:

```
<?php

$to = "someone@example.com";
$subject = "Test mail";
$message = "Hello! This is a simple email message.";
$from = "someonelse@example.com";
$headers = "From: $from";
mail($to,$subject,$message,$headers);
echo "Mail Sent.";

?>
```

در یک صفحه وب باید فرم مخصوصی به منظور دریافت پارامترهای نامه الکترونیکی و سپس ارسال آن طراحی شود. این فرم و عملکرد آن در مثال بعد بررسی شده است.

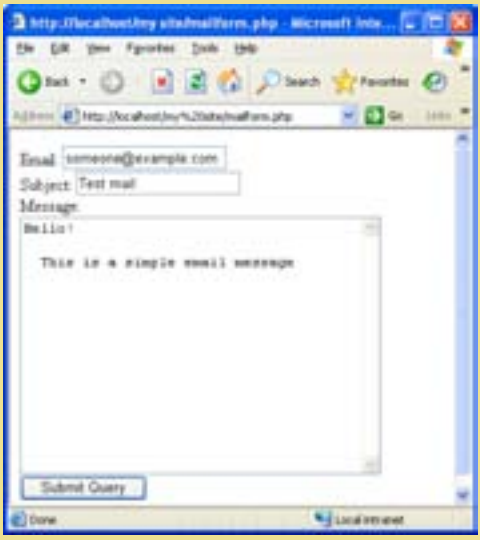

مثال:

```
<! 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>

<?php
if (isset($_REQUEST["email"]))
//if "email" is filled out, send email
  {
  //send email
  $email = $_REQUEST["email"] ;
  $subject = $_REQUEST["subject"] ;
  $message = $_REQUEST["message"] ;
  mail( "someone@example.com", "Subject: $subject",
 $message, "From: $email" );
  echo "Thank you for using our mail form";
  }
else
//if "email" is not filled out, display the form
  {
  echo "<form method="post" action="mailform.php">
  Email: <input name="email" type="text" /><br />
  Subject: <input name="subject" type="text" /><br />
  Message:<br />
  <textarea name="message"rows="15" cols="40">
  </textarea><br />
  <input type="submit"/>
  </form>";
  }
?>

</body>
</html>
```



شکل ۴-۵ طراحی فرم ارسال نامه الکترونیکی

چکیده فصل:

در PHP چنـد تابع درون سـاخته مهم وجـود دارند که امکان دسترسـی به متغیرهای متعددی را فراهم می‌کنند، این توابع عبارتند از:

$_POST

$_GET

$_REQUEST

اگر متدی که فرم از آن برای ارسال اطلاعات استفاده می‌کند، get باشد از تابع $_GET برای دسترسی به مقادیر موجود بر روی فرم استفاده می‌شود.

اگرمتد مورد استفاده فرم، post باشد از تابع $ _POST استفاده می‌شود.

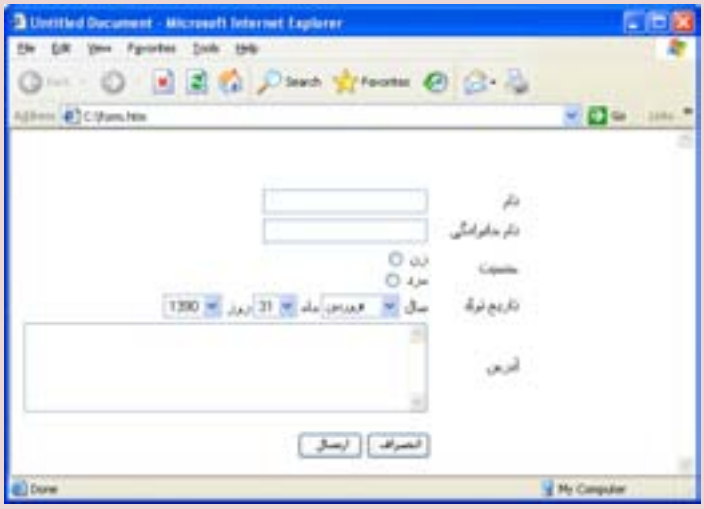
تابع $ _REQUEST می تواند صرف‌نظر از متد استفاده شده برای ارسال اطلاعات توسط فرم به مقادیر $_GET و $_POST و $ _COOKIE دسترسی داشته باشد.

توابع مهم و کاربردی فرم بارگذای عبارتند از:

- **$_FILES["file"]["error"]**:نتیجه خطای احتمالی، هنگام بارگذاری فایل را مشخص می کند. اگر هنگام بارگذاری فایل هیچ خطایی رخ نداده باشد، این تابع مقدار صفر را بر می گرداند.
- **$_ FILES ["file"]["name "]** : نام فایل بارگذاری شده را مشخص می‌کند.
- **$_ FILES ["file"]["type "]**: نوع و پسـوند فایل بارگذاری شـده را مشخص می‌کند. به عنوان مثال برای فایل‌های تصویری می‌تواند image/gif، image/tiff، image/jpg و ... باشد.
- **$_FILES["file"]["size"]**:اندازه فایل بارگذاری شده را بر حسب بایت مشخص می کند.
- **$_ FILES ["file"]["name _tmp"]** : مسیر ریشه فایل‌های ذخیره شده مربوط به سرویس دهنده را مشخص می‌کند.

خودآزمایی:

۱. یک فرم مطابق شـــکل زیر برای دریافت مشـــخصات کاربر طراحی کرده، اطلاعات آن را با اســـتفاده از متدpost برای صفحه‌ای با نام page2.php ارســـال کنید، ســـپس در فایل page2 .php با استفاده از تابع POST _$ اطلاعات وارد شده توسط کاربر را نمایش دهید.



شکل ۵-۵

۲. در سوال اول، اطلاعات وارد شده کاربر را با استفاده از تابع REQUEST _$نمایش دهید.

۳. به فرم مشـــخصات کاربر درشکل ۵-۵، امکان بارگذاری تصویری با قالب gif. را نیز اضافه کنید، سپس فایل بارگذاری شده را در پوشه ای با عنوان images ذخیره کنید.