

فصل ۵

طراحی برنامه‌های تحت وب و کار با فرم‌ها

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

۱- عملکرد متغیرهای درون ساخت `$_GET` ، `$_POST` و

`$_REQUEST` را شرح دهد.

۲- از `$_POST` ، `$_GET` و `$_REQUEST` در برنامه

استفاده کند.

۳- یک فرم برای بارگذاری فایل ایجاد کند.

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

کند.

مقدمه

در جلد اول این کتاب با فرم‌ها و بخش‌های آن آشنا شدید. همان‌طور که می‌دانید، یکی از مهم‌ترین مسائلی که هنگام طراحی وب سایت پویا لازم است در نظر گرفته شود، این است که اطلاعات باید به وسیله فرم‌ها از کاربر دریافت شوند و سپس به منظور تحلیل و پاسخگویی مناسب برای اسکریپت PHP ارسال شوند. دسترسی به اطلاعات ارسالی، توسط متغیرهای درون ساخت، به نام‌های GET-\$، post-\$ و REQUEST-\$ انجام می‌شود که در این فصل به بررسی آنها می‌پردازیم. معمولاً ساختار فرم و اجزای آن در فایل HTML ایجاد می‌شود. با ایجاد فرم و خصوصیت‌های مهم آن مثل action و rethed در جلد اول آشنا شدیم و در ادامه موارد کاربرد آنها را در این فصل فرامی‌گیریم. بهتر است برنامه‌ها را در محیط DreamWeaver نوشته و از امکانات آن استفاده نماییم.

۱-۵- POST-\$

نمونه کد زیر در فایل HTML به نام form.htm یک فرم با دو جعبه متن و یک دکمه submit برای ارسال اطلاعات ایجاد می‌کند.

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>

<form action="welcome.php" method="post">
Name: <input type="text" name="fname" />
<p>
Age: <input type="text" name="age" />
</p>
<p>
```

```
<input type="submit" Value = " Submit Query" />
```

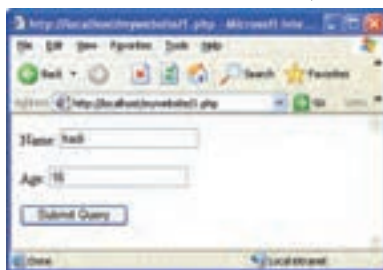
```
</p>
```

```
</form>
```

```
</body>
```

```
</html>
```

این فایل نام و سن کاربر را دریافت، و توسط متد post آن را برای فایل welcome.php ارسال می‌کند. خروجی فایل form.htm به صورت زیر خواهد بود. فرض کنید کاربر، اطلاعات فرم را مانند شکل زیر پر کرده باشد.



شکل ۵-۱

در فایل welcome.php با استفاده از متغیر درون ساخته `$_POST` آرگومان‌های ارسال شده توسط کاربر در فرم برنامه form.htm، 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.

۲-۵-GET - \$

اگر متدی که فرم از آن برای ارسال اطلاعات استفاده می‌کند، get باشد، متغیر `$_GET`

می‌تواند عملکردی مشابه متغیر `$_POST` در مثال فوق داشته باشد و برای دسترسی به مقادیر ارسال شده توسط فرم به کار برده شود.

فرض کنید در فایل `form.htm` از متد `get` برای ارسال اطلاعات استفاده شده 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!
```

۳-۵-\$_REQUEST

PHP یک متغیر 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 متغیر می‌تواند با هر دو متد ارسال `get` و `post` استفاده شود.
اگر فایل `welcome.php` را در مثال قبل، با استفاده از `$_REQUEST` بنویسیم، 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>
```

۴-۵-ایجاد فرم بارگذاری فایل

برای بارگذاری یک فایل در اینترنت، 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 (FileField) قرار دهید :

```
<input type="file" name="file" id="file" enctype="form-data/multipart" />
```

در این صورت دکمه Browse نیز روی فرم ظاهر می شود که امکان انتخاب فایل از طریق کادر محاوره Choose file را فراهم می کند.

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

مثال:

کد زیر یک فرم انتخاب فایل برای بارگذاری را ایجاد می کند :

```
<!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 />
```

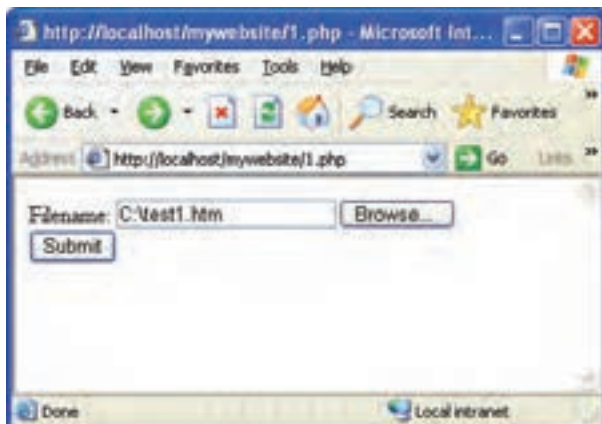
```
<input type="submit" name="submit" value="Submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

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



شکل ۲-۵

در این کد، فایل دریافت شده، برای اسکریپت `upload_file.php` ارسال می‌شود. اطلاعات فایل بارگذاری شده، از طریق متغیر از پیش تعریف شده `$_FILES` به اسکریپت `upload_file.php` ارسال می‌شود که به شکل‌های زیر می‌توان آن را به کار برد.

- `$_FILES["file"]["error"]`: نتیجه خطای احتمالی، هنگام بارگذاری فایل را مشخص می‌کند. اگر هنگام بارگذاری فایل هیچ خطایی رخ نداده باشد، این تابع مقدار صفر را بر می‌گرداند.
- `$_FILES["file"]["name"]`: نام فایل بارگذاری شده را مشخص می‌کند.
- `$_FILES["file"]["type"]`: نوع و پسوند فایل بارگذاری شده را مشخص می‌کند. به عنوان مثال برای فایل‌های تصویری می‌تواند `image/gif`، `image/tiff`، `image/jpg` و `image/jpeg`... باشد.
- `$_FILES["file"]["size"]`: اندازه فایل بارگذاری شده را برحسب بایت مشخص می‌کند.
- `$_FILES["file"]["tmp_name"]`: مسیر موقتی فایل‌های ذخیره شده مربوط به سرویس‌دهنده را مشخص می‌کند. تا زمانی که فایل‌ها توسط سرویس‌دهنده WAMP اجرا می‌شوند، این فایل‌ها در مسیر نصب سرویس‌دهنده WAMP ذخیره می‌شوند.

به منظور پیاده‌سازی اسکریپت `upload_file.php` می‌توانید کد زیر 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

```

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.

نکته : دقت کنید که آرگومان اول \$FILES مقدار خصوصیت name عنصر input از نوع فایل (File Field) است.

مثال فوق شکل بسیار ساده 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 اش را از نوع gif، .bmp. و یا .jpg با اندازه کمتر در ۲۰ کیلوبایت (حدوداً ۲۰۰۰۰ بایت) را انتخاب کند. سپس در کد اسکریپت upload_file.php نوع و اندازه آن را بررسی کنید و در صورت تناقض به فایل اجازه بارگذاری بر روی سرور دهنده را 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")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/bmp")))
&& ($_FILES["file"]["size"] <= 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Error " ;
}
else
{
echo "name" . $_FILES["file"]["name"] . "<br />";
echo "Size" . $_FILES["file"]["size"] . "<br />";
echo "Type" . $_FILES["file"]["type"] . "<br />";
echo "Stored in" . $_FILES["file"]["tmp_name"];
}
}
else
{
echo "invalid type or size";
}
?>

```

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 انجام شود.

برای انتقال فایل‌ها از ریشه، تابع `move_uploaded_file` مورد استفاده قرار می‌گیرد. شکل کلی تابع:

`move_uploaded_file` (آدرس جدید و آدرس مسیر موقتی)

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 بر روی 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. آرگومان ورودی این تابع مسیر کامل و نام فایل موردنظر است و خروجی آن `boolean` می‌باشد. کاربرد آن را در مثال بعد خواهید دید.

توصیه: بهتر است قبل از انجام عمل انتقال و نوشتن اسکریپت مربوط به آن، در مسیر ذخیره وبسایت خود، پوشه‌ای را برای ذخیره فایل‌های بارگذاری شده، ایجاد کنید. (در مثال فوق پوشه `upload`)

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 قبلاً آن را در مسیر ذخیره فایل‌های وبسایت، ایجاد کرده‌ایم، انجام می‌شود.

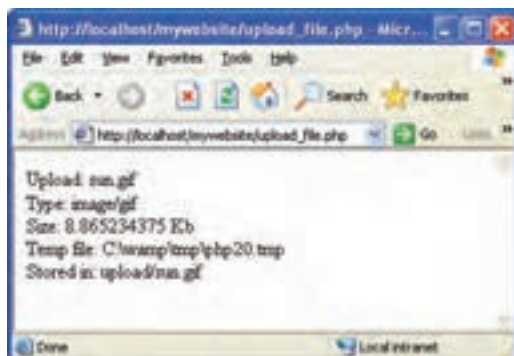
```
<?php
if(($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpg")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/bmp"))
&&($_FILES["file"]["size"] <= 20000))
```

```

{
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 gif . sun نشان می‌دهد.



شکل ۳-۵

مطالعه آزاد

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 می‌شود. این پارامتر می‌تواند آدرس پست الکترونیکی چند نفر را دریافت کند.
subject	این پارامتر عنوان نامه را تعیین می‌کند.
message	متن اصلی نامه را معین می‌کند و می‌تواند حداکثر تا ۷۰ کاراکتر را دریافت کند.
headers	این پارامتر اختیاری است و بخش‌های آدرس پست الکترونیکی فرستنده نامه، Cc و Bcc را به نامه الکترونیکی اضافه می‌کند.
parameters	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.

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 kerio mail server, MDaemon 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 = "someoneelse@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.

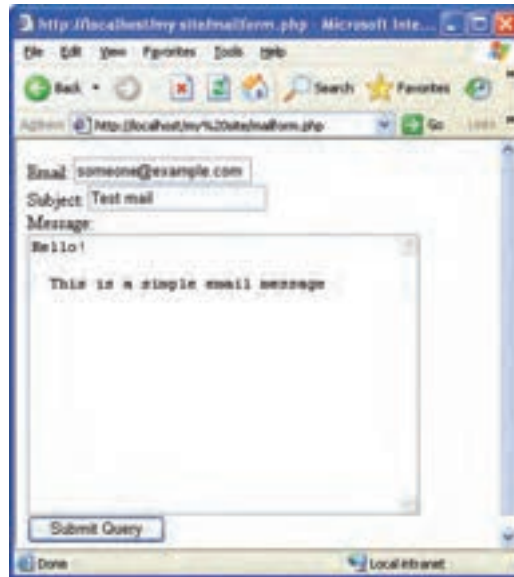
مثال:

```
<!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

در PHP چند متغیر درون ساخته مهم وجود دارند که امکان دسترسی به مقادیر ارسالی در فرم را فراهم می‌کنند. این توابع عبارتند از :

`$_POST`

`$_GET`

`$_REQUEST`

اگر متدی که فرم از آن برای ارسال اطلاعات استفاده می‌کند، `get` باشد از `$_GET` برای دسترسی به مقادیر ارسالی توسط فرم استفاده می‌شود.

اگر متد مورد استفاده فرم، `post` باشد از `$_POST` استفاده می‌شود.

`$_REQUEST` می‌تواند صرف‌نظر از متد استفاده شده برای ارسال اطلاعات، به مقادیر

`$_GET` و `$_POST` و `$_COOKIE` دسترسی داشته باشد.

مهم و کاربردی فرم بارگذاری `$_FILES` نام دارد که حاوی اطلاعات زیر است :

• `$_FILES["file"]["error"]` : نتیجه خطای احتمالی، هنگام بارگذاری فایل را مشخص می‌کند.

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

• `$_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 می‌تواند `image/gif`، `image/tiff`، `image/jpg` و ... باشد.

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

مشخص می‌کند.

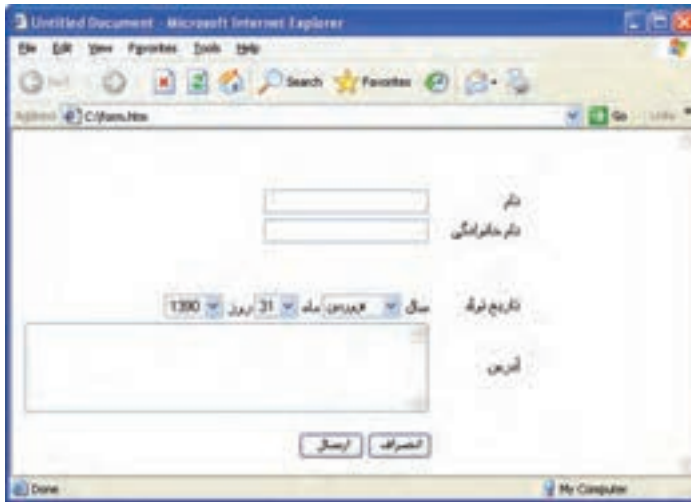
• `$_FILES["file"]["tmp_name"]` : مسیر ریشه فایل‌های ذخیره شده مربوط به

سرویس‌دهنده را مشخص می‌کند.

برای انتقال فایل از مسیر و نام موقتی به مسیر و نام جدید از تابع `move_uploaded_file`

استفاده می‌کنیم.

۱- یک فرم مطابق شکل زیر برای دریافت مشخصات کاربر طراحی کرده، اطلاعات آن را با استفاده از متد post برای صفحه‌ای با نام page2.php ارسال کنید، سپس در فایل page2.php با استفاده از تابع \$_POST اطلاعات وارد شده توسط کاربر را نمایش دهید.



شکل ۵-۵

۲- در سؤال اول، اطلاعات وارد شده کاربر را با استفاده از تابع \$_REQUEST نمایش دهید.

۳- به فرم مشخصات کاربر در شکل ۵-۵، امکان بارگذاری تصویری با قالب gif. را نیز اضافه کنید، سپس فایل بارگذاری شده را در پوشه‌ای با عنوان images ذخیره کنید.