

CSCI 241

Building an Online System

Department of CS & QM
Winthrop University
 Fall 2011

Prepared by X. Wang

Topics

- Building an Online System
- User Interface
- Coding for Reusability

Prepared by X. Wang

2

Building an Online System

- Our objective for learning PHP is to develop a database driven Web application (a system)
- So far we have introduced basic PHP programming skills and components for developing different individual operations in a web application
- Now we need to consider how to use these skills and components together to develop a complete web application, for example, our online bookstore

Prepared by X. Wang

3

Building an Online System

- Developing a complete web application is different from implementing a individual operation
- We have many new factors need to consider
- For example, we need to consider the user interface for the whole application, and consider how to reuse existing code for efficiently developing and maintaining the web site

Prepared by X. Wang

4

User Interface

- The user interface of an application should be friendly. Otherwise it is hard to get your users to return
- We have many hints to design and develop the user interface for a web application. Here let's consider page layout and navigation
- **Page Layout** is the placement of visual content items within a page. In a website, page layout should be consistent for certain pages
- **Page Navigation** is the access manner to all pages within a website.

Prepared by X. Wang

5

User Interface

- As a server-side programming language, PHP allows us not only to dynamically generate page contents, but also to dynamically control the page layout and page navigation (special page contents)
- For example, in a web application, the navigation links in the same page may be changing based on current operation states like sessions

Prepared by X. Wang

6

User Interface

- Consider the online bookstore, the "Register" and "Login" operations are not needed for a logged-in customer
- This means that once a customer logs in to the system, the navigation links in the coming pages should not display the Register and Login links anymore
- The next two snapshots show the differences about the user interface (navigation links) between before-login and after-login

Prepared by X. Wang

7

User Interface

- Before the login,



Prepared by X. Wang

8

User Interface

- After the login,



Prepared by X. Wang

9

User Interface

- Furthermore, this change should be extended to all PHP pages (generating user interface) that the logged-in customer can access
- For example, the Browse operation (only has one PHP page – booklist.php) can be accessed by a general user and a logged-in customer
- So in booklist.php, the PHP code should be able to generate these two navigations based on whether the user is a logged-in customer or not

Prepared by X. Wang

10

User Interface

- The Browse page before the login,



Prepared by X. Wang

11

User Interface

- The Browse page after the login,



Prepared by X. Wang

12

Coding for Reusability

- In the `booklist.php` page, we can see that the code segment in black color on the previous two slides generates the dynamic content for the `Browse` page
- The code before the segment is used to control page layout and navigation links that are common for other pages having the same user interface. We can write the code into a `header.php` file for other pages to call
- With the same reason, we can write the code after the black segment into a `footer.php` file for other pages to use too
- This is [Coding for Reusability](#)

Prepared by X. Wang

19

Coding for Reusability

- The following is the code for the `header.php` file

```
<?php session_start(); ?>
<!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" xml:lang="en"
    lang="en">
<head><title>Wang's Online Bookstore</title></head>
<body>
<table width="700" border="0" cellpadding="0" cellspacing="0"
    align="center" >
    <tr>
        <td height="61" bgcolor="#EEEEEE">
            .gif" width="700"
                height="61" border="0" />
            </td>
        </tr>
    </table>
```

Prepared by X. Wang

20

Coding for Reusability

```
<tr>
<td height="30" align="center" bgcolor="#11EE11">
<a href="index.php">Home</a> &nbsp;&nbsp;&nbsp;
<a href="booklist.php">Browse</a> &nbsp;&nbsp;&nbsp;
<a href="search.php">Search</a> &nbsp;&nbsp;&nbsp;
<a href="cartlist.php">Shopping Cart</a> &nbsp;&nbsp;&nbsp;
<?php
if (!isset($_SESSION["cust_id"])) {
print '<a href="register.php">Register</a> &nbsp;&nbsp;&nbsp; ';
print '<a href="login.php">Login</a>';
} else {
print '<a href="showorders.php">My Order</a> &nbsp;&nbsp;&nbsp; ';
print '<a href="logout.php">Logout</a>';
}
?>
</td>
</tr>
```

Prepared by X. Wang

21

Coding for Reusability

```
<?php
if (isset($_SESSION["cust_id"]))
{ print '<tr><td height="20" align="center" bgcolor="#11EEDD">';
print '<a href="profile.php">My Profile</a> &nbsp;&nbsp;&nbsp;';
print '<a href="chgnpassword.php">Password Change</a> &nbsp;&nbsp;&nbsp;';
print '<a href="chgname.php">Name Change</a> &nbsp;&nbsp;&nbsp;';
print '<a href="chgadress.php">Address Change</a>';
print '</td></tr>';
}
?>
<tr><td height="20" bgcolor="#EEEEEE" valign="top"align="right">
<?php
if (isset($_SESSION["lastname"])) {
print $_SESSION["lastname"] . ", ";
if (isset($_SESSION["firstname"]))
print $_SESSION["firstname"];
}
?>&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr><td height="300" bgcolor="#EEEEEE" valign="top">
```

Prepared by X. Wang

22

Coding for Reusability

- The following is the code for the `footer.php` file

```
&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
</body>
</html>
```

Prepared by X. Wang

23

Coding for Reusability

- With this code reorganization, if a PHP page uses the same user interface, we can directly include the `header.php` and `footer.php` into the page
- This also makes the code maintenance be easy. If we want to change the page layout, we only need to modify these two files
- To include a PHP file into another PHP file, we can use the directive `require_once`
- The `require_once` directive is similar to the `require` directive, but it only include once

Prepared by X. Wang

24

Coding for Reusability

- The following is the code for the new `booklist.php` file

```
<?php
require_once 'header.php';
require "db.php";
if (!$connection = @ mysql_connect($hostname, $dbusername,
$dbpassword))
    print("Could not connect to database");
elseif (!(@ mysql_select_db($databasename, $connection))
    showerror();
else {
    $query = "SELECT * FROM books ORDER BY ISBN";
    if (!($result = @ mysql_query ($query, $connection)))
        showerror();
    else {
        $num_result = mysql_num_rows($result);
        if ($num_result == 0)
            print("<p><br /> Sorry, no book in the bookstore!</p>");
    }
}
```

Prepared by X. Wang

25

Coding for Reusability

```
else {
    $one = " is";
    if ($num_result > 1) $one ="s are";
    print("{ $num_result } book{$one} found in the bookstore<br />
(The following list is ordered by ISBN-10)<br /><br />");

    for ($i=1; $i<=$num_result; $i++) {
        $row = @ mysql_fetch_array($result);
        print "$i. <a href='\"select.php?isbn={$row['ISBN']}\"'\>
{$row['title']}</a><br />";
        print "Author: {$row['authors']}<br />";
        print "ISBN-10: {$row['ISBN10']}<br />";
        print "ISBN-13: {$row['ISBN13']}<br />";
        print "Price: \"\${$row['price']}<br />";
        print "<br />";
    }
}
require_once 'footer.php';
?>
```

Prepared by X. Wang

26

Coding for Reusability

- If a page only contains HTML code besides the common user interface, we also write a PHP page to include `header.php` and `footer.php` directly with HTML code
- For example, the `Home` page of our online bookstore may only contain HTML contents
- The following is the code for the `home.php` page

Prepared by X. Wang

27

Coding for Reusability

```
<?php
require_once 'header.php';
?>
<h1>Bestseller books</h1>
<table>
<tr><td height="165">
    <a href="select.php?isbn=0764579665">
    </a> &nbsp;
    <b>Beginning PHP5, Apache, MySQL Web Development</b>
</td></tr>
<tr><td height="165">
    <a href="select.php?isbn=0132222205">
    </a> &nbsp;
    <b>Java How to Program, Seventh Edition</b>
</td></tr>
</table>
<?php
require_once 'footer.php';
?>
```

Prepared by X. Wang

28

Page Layout Control

- In the previous examples, we used `<table>` `</table>` element to control the page layout
- Actually, it is better to use `<div>` `</div>` element with CSS to control the page layout for all `include` files (e.g. `header.php` and `footer.php`) for code reusability
- With the `<div>` `</div>` element, we can also write PHP scripts in proper and separate `<div>` `</div>` blocks

Prepared by X. Wang

29