

```
1: <?
2:  /**
3:   * login5.php
4:   *
5:   * A simple login module that checks a username and password
6:   * against a MySQL table with no encryption.
7:   *
8:   * David J. Malan
9:   * Computer Science E-75
10:  * Harvard Extension School
11:  */
12:
13:  // enable sessions
14:  session_start();
15:
16:  // connect to database
17:  if (($connection = mysql_connect("", "", "")) === FALSE)
18:      die("Could not connect to database");
19:
20:  // select database
21:  if (mysql_select_db("", $connection) === FALSE)
22:      die("Could not select database");
23:
24:  // if username and password were submitted, check them
25:  if (isset($_POST["user"]) && isset($_POST["pass"]))
26:  {
27:      // prepare SQL
28:      $sql = sprintf("SELECT * FROM users WHERE user='%s'",
29:                    mysql_real_escape_string($_POST["user"]));
30:
31:      // execute query
32:      $result = mysql_query($sql);
33:      if ($result === FALSE)
34:          die("Could not query database");
35:
36:      // check whether we found a row
37:      if (mysql_num_rows($result) == 1)
38:      {
39:          // fetch row
40:          $row = mysql_fetch_assoc($result);
41:
42:          // check password
43:          if ($row["pass"] == $_POST["pass"])
44:          {
45:              // remember that user's logged in
46:              $_SESSION["authenticated"] = TRUE;
47:
```

```
48:                // redirect user to home page, using absolute path, per
49:                // http://us2.php.net/manual/en/function.header.php
50:                $host = $_SERVER["HTTP_HOST"];
51:                $path = rtrim(dirname($_SERVER["PHP_SELF"]), "\\");
52:                header("Location: http://$host$path/home.php");
53:                exit;
54:            }
55:        }
56:    }
57: ?>
58:
59: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
60:
61: <html xmlns="http://www.w3.org/1999/xhtml">
62:     <head>
63:         <title>Log In</title>
64:     </head>
65:     <body>
66:         <form action="<? echo $_SERVER["PHP_SELF"]; ?>" method="post">
67:             <table>
68:                 <tr>
69:                     <td>Username:</td>
70:                     <td>
71:                         <input name="user" type="text" /></td>
72:                 </tr>
73:                 <tr>
74:                     <td>Password:</td>
75:                     <td><input name="pass" type="password" /></td>
76:                 </tr>
77:                 <tr>
78:                     <td></td>
79:                     <td><input type="submit" value="Log In" /></td>
80:                 </tr>
81:             </table>
82:         </form>
83:     </body>
84: </html>
```

```
1: <?
2:  /**
3:   * login6.php
4:   *
5:   * A simple login module that checks a username and password
6:   * against a MySQL table with no encryption by asking for a binary answer.
7:   *
8:   * David J. Malan
9:   * Computer Science E-75
10:  * Harvard Extension School
11:  */
12:
13:  // enable sessions
14:  session_start();
15:
16:  // connect to database
17:  if (($connection = mysql_connect("", "", "")) === FALSE)
18:      die("Could not connect to database");
19:
20:  // select database
21:  if (mysql_select_db("", $connection) === FALSE)
22:      die("Could not select database");
23:
24:  // if username and password were submitted, check them
25:  if (isset($_POST["user"]) && isset($_POST["pass"]))
26:  {
27:      // prepare SQL
28:      $sql = sprintf("SELECT 1 FROM users WHERE user='%s' AND pass='%s'",
29:                    mysql_real_escape_string($_POST["user"]),
30:                    mysql_real_escape_string($_POST["pass"]));
31:
32:      // execute query
33:      $result = mysql_query($sql);
34:      if ($result === FALSE)
35:          die("Could not query database");
36:
37:      // check whether we found a row
38:      if (mysql_num_rows($result) == 1)
39:      {
40:          // remember that user's logged in
41:          $_SESSION["authenticated"] = TRUE;
42:
43:          // redirect user to home page, using absolute path, per
44:          // http://us2.php.net/manual/en/function.header.php
45:          $host = $_SERVER["HTTP_HOST"];
46:          $path = rtrim(dirname($_SERVER["PHP_SELF"]), "\\");
47:          header("Location: http://$host$path/home.php");
```

```
48:         exit;
49:     }
50: }
51: ?>
52:
53: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
54:
55: <html xmlns="http://www.w3.org/1999/xhtml">
56:     <head>
57:         <title>Log In</title>
58:     </head>
59:     <body>
60:         <form action="<? echo $_SERVER["PHP_SELF"]; ?>" method="post">
61:             <table>
62:                 <tr>
63:                     <td>Username:</td>
64:                     <td>
65:                         <input name="user" type="text" /></td>
66:                 </tr>
67:                 <tr>
68:                     <td>Password:</td>
69:                     <td><input name="pass" type="password" /></td>
70:                 </tr>
71:                 <tr>
72:                     <td></td>
73:                     <td><input type="submit" value="Log In" /></td>
74:                 </tr>
75:             </table>
76:         </form>
77:     </body>
78: </html>
```

```
1: <?
2: /**
3:  * login7.php
4:  *
5:  * A simple login module that checks a username and password
6:  * against a MySQL table with weak encryption (well, a weak hash).
7:  *
8:  * David J. Malan
9:  * Computer Science E-75
10:  * Harvard Extension School
11:  */
12:
13: // enable sessions
14: session_start();
15:
16: // connect to database
17: if (($connection = mysql_connect("", "", "")) === FALSE)
18:     die("Could not connect to database");
19:
20: // select database
21: if (mysql_select_db("", $connection) === FALSE)
22:     die("Could not select database");
23:
24: // if username and password were submitted, check them
25: if (isset($_POST["user"]) && isset($_POST["pass"]))
26: {
27:     // prepare SQL
28:     $sql = sprintf("SELECT 1 FROM users WHERE user='%s' AND pass=PASSWORD('%s')",
29:                    mysql_real_escape_string($_POST["user"]),
30:                    mysql_real_escape_string($_POST["pass"]));
31:
32:     // execute query
33:     $result = mysql_query($sql);
34:     if ($result === FALSE)
35:         die("Could not query database");
36:
37:     // check whether we found a row
38:     if (mysql_num_rows($result) == 1)
39:     {
40:         // remember that user's logged in
41:         $_SESSION["authenticated"] = TRUE;
42:
43:         // redirect user to home page, using absolute path, per
44:         // http://us2.php.net/manual/en/function.header.php
45:         $host = $_SERVER["HTTP_HOST"];
46:         $path = rtrim(dirname($_SERVER["PHP_SELF"]), "\\");
47:         header("Location: http://$host$path/home.php");
```

```
48:         exit;
49:     }
50: }
51: ?>
52:
53: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
54:
55: <html xmlns="http://www.w3.org/1999/xhtml">
56:     <head>
57:         <title>Log In</title>
58:     </head>
59:     <body>
60:         <form action="<? echo $_SERVER["PHP_SELF"]; ?>" method="post">
61:             <table>
62:                 <tr>
63:                     <td>Username:</td>
64:                     <td>
65:                         <input name="user" type="text" /></td>
66:                 </tr>
67:                 <tr>
68:                     <td>Password:</td>
69:                     <td><input name="pass" type="password" /></td>
70:                 </tr>
71:                 <tr>
72:                     <td></td>
73:                     <td><input type="submit" value="Log In" /></td>
74:                 </tr>
75:             </table>
76:         </form>
77:     </body>
78: </html>
```

```
1: <?
2:  /**
3:   * login8.php
4:   *
5:   * A simple login module that checks a username and password
6:   * against a MySQL table with strong encryption (but insecure secret).
7:   *
8:   * David J. Malan
9:   * Computer Science E-75
10:  * Harvard Extension School
11:  */
12:
13:  // enable sessions
14:  session_start();
15:
16:  // connect to database
17:  if (($connection = mysql_connect("", "", "")) === FALSE)
18:      die("Could not connect to database");
19:
20:  // select database
21:  if (mysql_select_db("", $connection) === FALSE)
22:      die("Could not select database");
23:
24:  // if username and password were submitted, check them
25:  if (isset($_POST["user"]) && isset($_POST["pass"]))
26:  {
27:      // prepare SQL
28:      $sql = sprintf("SELECT 1 FROM users WHERE user='%s' AND pass=AES_ENCRYPT('%s', 'secret')",
29:                    mysql_real_escape_string($_POST["user"]),
30:                    mysql_real_escape_string($_POST["pass"]));
31:
32:      // execute query
33:      $result = mysql_query($sql);
34:      if ($result === FALSE)
35:          die("Could not query database");
36:
37:      // check whether we found a row
38:      if (mysql_num_rows($result) == 1)
39:      {
40:          // remember that user's logged in
41:          $_SESSION["authenticated"] = TRUE;
42:
43:          // redirect user to home page, using absolute path, per
44:          // http://us2.php.net/manual/en/function.header.php
45:          $host = $_SERVER["HTTP_HOST"];
46:          $path = rtrim(dirname($_SERVER["PHP_SELF"]), "\\");
47:          header("Location: http://$host$path/home.php");
```

```
48:         exit;
49:     }
50: }
51: ?>
52:
53: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
54:
55: <html xmlns="http://www.w3.org/1999/xhtml">
56:     <head>
57:         <title>Log In</title>
58:     </head>
59:     <body>
60:         <form action="<? echo $_SERVER["PHP_SELF"]; ?>" method="post">
61:             <table>
62:                 <tr>
63:                     <td>Username:</td>
64:                     <td>
65:                         <input name="user" type="text" /></td>
66:                 </tr>
67:                 <tr>
68:                     <td>Password:</td>
69:                     <td><input name="pass" type="password" /></td>
70:                 </tr>
71:                 <tr>
72:                     <td></td>
73:                     <td><input type="submit" value="Log In" /></td>
74:                 </tr>
75:             </table>
76:         </form>
77:     </body>
78: </html>
```



```
1:
2: <!DOCTYPE html PUBLIC
3:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
4:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
5:
6: <html xmlns="http://www.w3.org/1999/xhtml">
7:   <head>
8:     <title>Lolcat of teh Day</title>
9:   </head>
10:  <body>
11:    <div align="center" style="padding: 20px;">
12:      <h1>Lolcat of teh Day</h1>
13:      <?
14:
15:          $xml = new SimpleXMLElement(file_get_contents("http://feedproxy.google.com/ICanHasCheezburger?format=xml"
16: ));
17:          $item = $xml->channel->item[0];
18:          preg_match("/^.* - (.*)</", $item->description, $matches);
19:          $alt = htmlspecialchars($matches[1], ENT_QUOTES);
20:          $link = $item->link;
21:          foreach ($item->children("http://search.yahoo.com/mrss/") as $content)
22:          {
23:              $attributes = $content->attributes();
24:              $src = $attributes["url"];
25:          }
26:          print("<a href='{ $link }'><img alt='{ $alt }' border='0' src='{ $src }' /></a>");
27:
28:      </div>
29:  </body>
30: </html>
31:
```

```
1: <?
2:
3:     // ensure complete form was submitted
4:     if (!isset($_POST["name"]) || !isset($_POST["item"]))
5:     {
6:         header("Location: http://www.cs75.net/lectures/4/src/lunch/lunch.php");
7:         exit;
8:     }
9:
10:    // open CSV file for appending
11:    $handle = fopen("orders.csv", "a");
12:
13:    // acquire exclusive lock
14:    flock($handle, LOCK_EX);
15:
16:    // add order to CSV file
17:    $order = array($_POST["name"], $_POST["item"]);
18:    fputcsv($handle, $order);
19:    fclose($handle);
20:
21: ?>
22:
23: <!DOCTYPE html PUBLIC
24:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
25:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
26:
27: <html xmlns="http://www.w3.org/1999/xhtml">
28:     <head>
29:         <title>Lunch</title>
30:     </head>
31:     <body>
32:         One <?= $_POST["item"] ?> for <?= $_POST["name"] ?>, coming right up!
33:     </body>
34: </html>
```

```
1: <?
2:     $xml = new SimpleXMLElement(file_get_contents("menu.xml"));
3: ?>
4:
5: <!DOCTYPE html PUBLIC
6:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
7:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
8:
9: <html xmlns="http://www.w3.org/1999/xhtml">
10:     <head>
11:         <title>Lunch</title>
12:     </head>
13:     <body>
14:         <form action="csv.php" method="post">
15:             <b>Name:</b> <input name="name" type="text" /> <input type="submit" value="Submit Order" />
16:             <br /><br />
17:             <table border="0">
18:                 <? foreach ($xml->xpath("/menu/category[@name='Specialty Sandwiches']/item") as $item): ?>
19:                     <tr>
20:                         <td valign="top"><input id="<?= $item["name"] ?>" name="item" type="radio" value="<?= $item["name"] ?>"
/></td>
21:                         <td>
22:                             <label for="<?= $item["name"] ?>">
23:                                 <b><?= $item["name"] ?></b>
24:                                 <br />
25:                                 <?= $item ?>
26:                             </label>
27:                         </td>
28:                     </tr>
29:                 <? endforeach ?>
30:             </table>
31:         </form>
32:     </body>
33: </html>
```

```
1: <menu>
2:   <fillings>
3:     <filling>famous chicken salad</filling>
4:     <filling>turkey</filling>
5:     <filling>grilled chicken breast</filling>
6:     <filling>roast beef</filling>
7:     <filling>smoked Virginia ham</filling>
8:     <filling>albacore tuna salad</filling>
9:     <filling>egg salad</filling>
10:    <filling>three cheese</filling>
11:    <filling>vegetarian</filling>
12:  </fillings>
13:  <breads>
14:    <bread>sourdough</bread>
15:    <bread>marble rye</bread>
16:    <bread>French baguette</bread>
17:    <bread>onion roll</bread>
18:    <bread>bulkie roll</bread>
19:    <bread>country white</bread>
20:    <bread>multi-grain</bread>
21:    <bread>pita pocket</bread>
22:    <bread>soft tortilla roll-up</bread>
23:  </breads>
24:  <cheeses>
25:    <cheese>Swiss</cheese>
26:    <cheese>cheddar</cheese>
27:    <cheese>low fat cheddar</cheese>
28:    <cheese>American</cheese>
29:    <cheese>mozzarella</cheese>
30:    <cheese>feta</cheese>
31:    <cheese>dill havarti</cheese>
32:    <cheese>provolone</cheese>
33:    <cheese>fresh mozzarella</cheese>
34:    <cheese>Monterey jack</cheese>
35:    <cheese>garlic herb spread</cheese>
36:    <cheese>brie</cheese>
37:    <cheese>low-fat cheese</cheese>
38:  </cheeses>
39:  <condiments>
40:    <condiment>lettuce</condiment>
41:    <condiment>tomato</condiment>
42:    <condiment>sliced red onion</condiment>
43:    <condiment>roasted red peppers</condiment>
44:    <condiment>cucumbers</condiment>
45:    <condiment>carrots</condiment>
46:    <condiment>caramelized onions</condiment>
47:  </condiments>
```

```
48: <spreads>
49:   <spread>mayonnaise</spread>
50:   <spread>mustard</spread>
51:   <spread>honey mustard</spread>
52:   <spread>Russian dressing</spread>
53:   <spread>guacamole</spread>
54:   <spread>chutney</spread>
55:   <spread>balsamic vinegar </spread>
56: </spreads>
57: <category color="f8981d" name="Entree Salads">
58:   <item name="Countryside Salad with chicken">
59:     <![CDATA[
60:       mixed greens, spiced pecans, dried cranberries and fresh goat cheese with balsamic vinaigrette
61:     ]]>
62:   </item>
63:   <item name="Countryside Salad with flank steak">
64:     <![CDATA[
65:       mixed greens, spiced pecans, dried cranberries and fresh goat cheese with balsamic vinaigrette
66:     ]]>
67:   </item>
68:   <item name="Fiesta Chicken Salad with tortilla chips">
69:     <![CDATA[
70:       romaine, black olives, corn, scallion, black beans, tomato, grilled chicken and a dollop of guacamole with
lite salsa-ranch dressing
71:     ]]>
72:   </item>
73:   <item name="Fiesta Chicken Salad without tortilla chips">
74:     <![CDATA[
75:       romaine, black olives, corn, scallion, black beans, tomato, grilled chicken and a dollop of guacamole with
lite salsa-ranch dressing
76:     ]]>
77:   </item>
78:   <item name="Waldorf Salad with Chicken">
79:     <![CDATA[
80:       romaine, green apples, seedless grapes, crumbled walnuts, grilled chicken and Waldorf dressing
81:     ]]>
82:   </item>
83:   <item name="Buffalo Chicken Salad">
84:     <![CDATA[
85:       breaded chicken tenders and spicy buffalo sauce, greens and blue cheese dressing
86:     ]]>
87:   </item>
88:   <item name="Florentino">
89:     <![CDATA[
90:       baby spinach, roasted red peppers, crumbled gorgonzola, caramelized onions, crispy bacon drizzled with bals
amic vinaigrette
91:     ]]>
```

```
92:      </item>
93:      <item name="Rebecca's Cobb Salad">
94:        <![CDATA[
95:          Boston lettuce topped with sliced grilled chicken breast, crumbled blue cheese, black olives and red pepper
s
96:        ]]>
97:      </item>
98:      <item name="Salad Topper">
99:        <![CDATA[
100:          house salad topped with grilled chicken, albacore tuna salad, chicken salad, deli meats or cheeses and choi
ce of dressing
101:        ]]>
102:      </item>
103:      <item name="Flank Steak Caesar Salad">
104:        <![CDATA[
105:          our Caesar salad topped with chef prepared flanksteak
106:        ]]>
107:      </item>
108:      <item name="Chicken Caesar Salad">
109:        <![CDATA[
110:          our Caesar salad topped with grilled sliced chicken breast
111:        ]]>
112:      </item>
113:      <item name="Market Salad">
114:        <![CDATA[
115:          ham, turkey, dill havarti and cheddar cheese fanned over garden greens with tomatoes, cucumbers and mushroo
ms and <strong>choice of dressing</strong>
116:        ]]>
117:      </item>
118:      <item name="Grilled Antipasto Misto Salad">
119:        <![CDATA[
120:          grilled zucchini, summer squash, eggplant, red and green peppers, onions, carrots, asparagus and portobello
mushrooms &#8212; all tossed in a roasted red pepper vinaigrette and served with stuffed mushrooms and roasted pineapple
121:        ]]>
122:      </item>
123:    </category>
124:    <category color="34B233" name="Specialty Sandwiches">
125:      <item name="Mediterranean Chicken">
126:        <![CDATA[
127:          grilled marinated chicken breast with hummus, red onion, roasted red pepper and lemon vinaigrette on rosema
ry focaccia
128:        ]]>
129:      </item>
130:      <item name="Smoked Turkey and Apple">
131:        <![CDATA[
132:          smoked turkey breast, thinly sliced granny smith apple, Vermont Cheddar and honey mustard on multi-grain br
ead
```

```
133:     ]]>
134:   </item>
135:   <item name="California Grilled Chicken Club">
136:     <![CDATA[
137:       marinated grilled chicken breast, guacamole, tomato, romaine lettuce, cheddar and cilantro mayonnaise on mu
liti-grain bread
138:     ]]>
139:   </item>
140:   <item name="Roasted Veggie Baguette">
141:     <![CDATA[
142:       roasted squash, zucchini, eggplant, tomato, onion, peppers, lettuce and garlic mayonnaise on French bread
143:     ]]>
144:   </item>
145:   <item name="Bello Bello Panini">
146:     <![CDATA[
147:       slow roasted portobello mushrooms, caramelized onions, fresh mozzarella, tomatoes, fresh basil and roasted
garlic mayonnaise on focaccia
148:     ]]>
149:   </item>
150:   <item name="Roasted Eggplant Panini">
151:     <![CDATA[
152:       tender eggplant, basil oil, roasted red peppers, caramelized onions, tomato and feta cheese on focaccia
153:     ]]>
154:   </item>
155:   <item name="Spicy Grilled Chicken">
156:     <![CDATA[
157:       grilled chicken with habanero hot sauce with lettuce, cucumber, peppers, swiss cheese and balsamic vinaigre
tte on a French baguette
158:     ]]>
159:   </item>
160:   <item name="Grilled Tuscan Chicken">
161:     <![CDATA[
162:       grilled chicken breast with spinach, roasted red pepper, fresh mozzarella, tomato, and Italian dressing on
focaccia
163:     ]]>
164:   </item>
165:   <item name="Chicken Caprese">
166:     <![CDATA[
167:       tomato, fresh mozzarella, fresh basil leaves and balsamic vinegar or vinaigrette on a French baguette
168:     ]]>
169:   </item>
170:   <item name="Greek Vegetarian">
171:     <![CDATA[
172:       sweet peppers, feta, cucumbers, lettuce, tomato, onions, and hummus on country white or French baguette
173:     ]]>
174:   </item>
175:   <item name="Italian Turkey">
```

```
176:      <![CDATA[
177:          smoked turkey, salami, provolone pepperoni, red onion, olives, lettuce, Italian dressing on focaccia
178:      ]]>
179:  </item>
180:  <item name="Tuscan Turkey">
181:      <![CDATA[
182:          smoked turkey, mozzarella, roasted peppers, wild greens and pesto mayonnaise on focaccia
183:      ]]>
184:  </item>
185:  <item name="Steakhouse Roast Beef">
186:      <![CDATA[
187:          roast beef, horseradish mayonnaise, caramelized onions, cheddar cheese on a French baguette
188:      ]]>
189:  </item>
190:  <item name="Zesty Roast Beef">
191:      <![CDATA[
192:          roast beef, Vermont cheddar, tomato, spinach, onions and dijon mustard on country white
193:      ]]>
194:  </item>
195:  <item name="Albacore Tuna Club">
196:      <![CDATA[
197:          tuna salad, Swiss cheese, bacon, lettuce and tomato on a croissant
198:      ]]>
199:  </item>
200: </category>
201: <category color="00ADD0" name="Specialty Wraps">
202:   <item name="Greek Chicken Wrap">
203:       <![CDATA[
204:           grilled chicken breast, lettuce, tomato, red onion, crumbled feta, roasted red pepper, cucumbers and olives
in a flavored tortilla
205:       ]]>
206:   </item>
207:   <item name="California Turkey Club Wrap">
208:       <![CDATA[
209:           smoked turkey breast, bacon, guacamole, salad mix and tomato with garlic mayonnaise in a plain tortilla
210:       ]]>
211:   </item>
212:   <item name="Cool Cheese Steak">
213:       <![CDATA[
214:           roast beef, Boursin cheese, roasted red peppers, salad mix, tomatoes and red onions in a plain tortilla
215:       ]]>
216:   </item>
217:   <item name="Buffalo Chicken Wrap">
218:       <![CDATA[
219:           grilled chicken breast, hot sauce, pepper jack, lettuce mix, tomato, red onion and blue cheese dressing in
a plain tortilla
220:       ]]>
```



```
221:     </item>
222:     <item name="Cobb Salad Wrap">
223:         <![CDATA[
224:             grilled chicken, bacon, guacamole, lettuce mix, tomato and red onion in a plain tortilla
225:         ]]>
226:     </item>
227:     <item name="Flank Steak Wrap">
228:         <![CDATA[
229:             thinly shaved flank steak, feta, olives and lettuce mix with greek dressing in a plain tortilla
230:         ]]>
231:     </item>
232:     <item name="Asian Chicken Wrap">
233:         <![CDATA[
234:             grilled chicken breast, roasted peppers and tomato with honey-ginger dressing in a plain tortilla
235:         ]]>
236:     </item>
237:     <item name="Southwest Turkey Wrap">
238:         <![CDATA[
239:             roast turkey, cheddar, lettuce, tomato, guacamole and chipotle mayonnaise in a tomato tortilla
240:         ]]>
241:     </item>
242:     <item name="Southwest Chicken Wrap">
243:         <![CDATA[
244:             grilled chicken, cheddar, lettuce, tomato, guacamole and chipotle mayonnaise in a tomato tortilla
245:         ]]>
246:     </item>
247:     <item name="Hawaiian Ham Wrap">
248:         <![CDATA[
249:             ham, pineapple slices, havarti cheese, tomato and wild greens with mango chutney on a plain tortilla
250:         ]]>
251:     </item>
252:     <item name="Teriyaki Chicken Wrap">
253:         <![CDATA[
254:             teriyaki chicken breast, lettuce, tomato and red onion, in a flavored tortilla
255:         ]]>
256:     </item>
257:     <item name="Scilliano Wrap">
258:         <![CDATA[
259:             mozzarella, roasted red peppers, sun dried tomato spread and basil
260:         ]]>
261:     </item>
262:     <item name="Hummus and Vegetable Wrap">
263:         <![CDATA[
264:             lettuce, roasted eggplant, hummus, feta, tomatoes and red peppers
265:         ]]>
266:     </item>
267:     <item name="Mixed Green Wrap">
```

```
268:      <![CDATA[
269:        wild mixed greens, tomatoes, goat cheese, spiced pecans, red onion and lite raspberry vinegrette
270:      ]]>
271:    </item>
272:    <item name="Chicken Caesar Wrap in a plain tortilla">
273:      <![CDATA[
274:        grilled chicken, parmesan, romaine, croutons, lettuce and Caesar dressing
275:      ]]>
276:    </item>
277:    <item name="Chicken Caesar Wrap in a pita">
278:      <![CDATA[
279:        grilled chicken, parmesan, romaine, croutons, lettuce and Caesar dressing
280:      ]]>
281:    </item>
282:    <item name="The Rebecca Roll">
283:      <![CDATA[
284:        smoked turkey breast, crisp bacon, chopped scallions, dill havarti cheese and cranberry - chipotle sauce wr
appd in a flour tortilla
285:      ]]>
286:    </item>
287:  </category>
288: </menu>
```

**03/02/09**  
**11:20:20**

lectures/5/src/lunch/development/orders.xml

1

```
1: <orders>  
2: </orders>
```

```
1: <?
2:
3:     // ensure complete form was submitted
4:     if (!isset($_POST["name"]) || !isset($_POST["item"]))
5:     {
6:         header("Location: http://www.cs75.net/lectures/4/src/lunch/lunch.php");
7:         exit;
8:     }
9:
10:    try
11:    {
12:        // open database
13:        $dbh = new PDO("sqlite:orders.db");
14:        $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
15:
16:        // prepare fields
17:        $name = $dbh->quote($_POST["name"]);
18:        $item = $dbh->quote($_POST["item"]);
19:
20:        // insert order
21:        $dbh->exec("INSERT INTO orders (name, item) VALUES($name, $item)");
22:    }
23:    catch (PDOException $e)
24:    {
25:        die($e->getMessage());
26:    }
27:
28: ?>
29:
30: <!DOCTYPE html PUBLIC
31:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
32:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
33:
34: <html xmlns="http://www.w3.org/1999/xhtml">
35:     <head>
36:         <title>Lunch</title>
37:     </head>
38:     <body>
39:         One <?= $_POST["item"] ?> for <?= $_POST["name"] ?>, coming right up!
40:     </body>
41: </html>
```

```
1: <?
2:
3:     // ensure complete form was submitted
4:     if (!isset($_POST["name"]) || !isset($_POST["item"]))
5:     {
6:         header("Location: http://www.cs75.net/lectures/4/src/lunch/lunch.php");
7:         exit;
8:     }
9:
10:    // open XML file for reading + writing
11:    $handle = fopen("orders.xml", "r+");
12:
13:    // acquire exclusive lock
14:    flock($handle, LOCK_EX);
15:
16:    // read contents of XML file
17:    $contents = fread($handle, filesize("orders.xml"));
18:
19:    // build DOM out of contents
20:    $xml = new SimpleXMLElement($contents);
21:
22:    // add order
23:    $order = $xml->addChild("order");
24:    $order->addChild("name", $_POST["name"]);
25:    $order->addChild("item", $_POST["item"]);
26:
27:    // overwrite original XML file
28:    rewind($handle);
29:    fwrite($handle, $xml->asXML());
30:    fclose($handle);
31: ?>
32:
33: <!DOCTYPE html PUBLIC
34:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
35:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
36:
37: <html xmlns="http://www.w3.org/1999/xhtml">
38:   <head>
39:     <title>Lunch</title>
40:   </head>
41:   <body>
42:     One <?= $_POST["item"] ?> for <?= $_POST["name"] ?>, coming right up!
43:   </body>
44: </html>
```