

```
1. #!/usr/bin/env php
2. <?php
3.
4.     require("../includes/config.php");
5.
6.     // select all rows from users table
7.     $rows = query("SELECT * FROM users");
8.
9.     // iterate over rows
10.    foreach ($rows as $row)
11.    {
12.        // print summary of row
13.        printf("Name is %s, number is %s, and carrier is %s.\n", $row["name"], $row["number"], $row["carrier"]);
14.
15.        // send personalized text
16.        text($row["number"], $row["carrier"], "hello, ${row["name"]}");
17.    }
18.
19.    ?>
```

```
1. CREATE TABLE IF NOT EXISTS `users` (  
2.   `id` int(11) NOT NULL AUTO_INCREMENT,  
3.   `name` varchar(64) NOT NULL,  
4.   `number` char(10) NOT NULL,  
5.   `carrier` enum('AT&T','Sprint','T-Mobile','Verizon') NOT NULL,  
6.   PRIMARY KEY (`id`)  
7. ) ENGINE=InnoDB AUTO_INCREMENT=1 ;
```

```
1. <?php
2.
3.     // display errors, warnings, and notices
4.     ini_set("display_errors", true);
5.     error_reporting(E_ALL);
6.
7.     // requirements
8.     require("constants.php");
9.     require("functions.php");
10.    require("libphp-phpmailer/class.phpmailer.php");
11.
12.    // enable sessions
13.    session_start();
14. ?>
```

```
1. <?php
2.
3.     // your database's name
4.     define("DATABASE", "lecture");
5.
6.     // your database's password
7.     define("PASSWORD", "crimson");
8.
9.     // your database's server
10.    define("SERVER", "localhost");
11.
12.    // your database's username
13.    define("USERNAME", "jharvard");
14.
15. ?>
```

```

1. <?php
2.
3.     require_once("config.php");
4.
5.     /**
6.      * Apologizes to user with message.
7.      */
8.     function apologize($message)
9.     {
10.         render("apology.php", ["message" => $message]);
11.         exit;
12.     }
13.
14.     /**
15.      * Executes SQL statement, possibly with parameters, returning
16.      * an array of all rows in result set or false on (non-fatal) error.
17.      */
18.     function query(/* $sql [, ... ] */)
19.     {
20.         // SQL statement
21.         $sql = func_get_arg(0);
22.
23.         // parameters, if any
24.         $parameters = array_slice(func_get_args(), 1);
25.
26.         // try to connect to database
27.         static $handle;
28.         if (!isset($handle))
29.         {
30.             try
31.             {
32.                 // connect to database
33.                 $handle = new PDO("mysql:dbname=" . DATABASE . ";host=" . SERVER, USERNAME, PASSWORD);
34.
35.                 // ensure that PDO::prepare returns false when passed invalid SQL
36.                 $handle->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
37.             }
38.             catch (Exception $e)
39.             {
40.                 // trigger (big, orange) error
41.                 trigger_error($e->getMessage(), E_USER_ERROR);
42.                 exit;
43.             }
44.         }
45.
46.         // prepare SQL statement
47.         $statement = $handle->prepare($sql);
48.         if ($statement === false)

```

```
49.     {
50.         // trigger (big, orange) error
51.         trigger_error($handle->errorInfo()[2], E_USER_ERROR);
52.         exit;
53.     }
54.
55.     // execute SQL statement
56.     $results = $statement->execute($parameters);
57.
58.     // return result set's rows, if any
59.     if ($results !== false)
60.     {
61.         return $statement->fetchAll(PDO::FETCH_ASSOC);
62.     }
63.     else
64.     {
65.         return false;
66.     }
67. }
68.
69. /**
70.  * Renders template, passing in values.
71.  */
72. function render($template, $values = [])
73. {
74.     // if template exists, render it
75.     if (file_exists("../templates/$template"))
76.     {
77.         // extract variables into local scope
78.         extract($values);
79.
80.         // render header
81.         require("../templates/header.php");
82.
83.         // render template
84.         require("../templates/$template");
85.
86.         // render footer
87.         require("../templates/footer.php");
88.     }
89.
90.     // else err
91.     else
92.     {
93.         trigger_error("Invalid template: $template", E_USER_ERROR);
94.     }
95. }
96.
```

```
97.  /**
98.   * Sends a text. Returns true on success, else false.
99.   */
100. function text($number, $carrier, $message)
101. {
102.     // determine address
103.     switch ($carrier)
104.     {
105.         case "AT&T":
106.             $address = "{ $number}@txt.att.net";
107.             break;
108.
109.         case "Sprint":
110.             $address = "{ $number}@messaging.sprintpcs.com";
111.             break;
112.
113.         case "T-Mobile":
114.             $address = "{ $number}@tmomail.net";
115.             break;
116.
117.         case "Verizon":
118.             $address = "{ $number}@vtext.com";
119.             break;
120.     }
121.     if (!isset($address))
122.     {
123.         return false;
124.     }
125.
126.     // instantiate mailer
127.     $mail = new PHPMailer();
128.
129.     // use SMTP
130.     $mail->IsSMTP();
131.     $mail->Host = "smtp.fas.harvard.edu";
132.     $mail->Port = 587;
133.     $mail->SMTPSecure = "tls";
134.
135.     // set From:
136.     $mail->SetFrom("jharvard@cs50.harvard.edu");
137.
138.     // set To:
139.     $mail->AddAddress($address);
140.
141.     // set body
142.     $mail->Body = $message;
143.
144.     // send text
```

```
145.         if ($mail->Send())
146.         {
147.             return true;
148.         }
149.         else
150.         {
151.             return false;
152.         }
153.     }
154.
155. ?>
```



```
1. <?php
2.
3.  /**
4.   * counter.php
5.   *
6.   * David J. Malan
7.   * malan@harvard.edu
8.   *
9.   * Implements a counter. Demonstrates sessions.
10.  */
11.
12.  // enable sessions
13.  session_start();
14.
15.  // check counter
16.  if (isset($_SESSION["counter"]))
17.  {
18.      $counter = $_SESSION["counter"];
19.  }
20.  else
21.  {
22.      $counter = 0;
23.  }
24.
25.  // increment counter
26.  $_SESSION["counter"] = $counter + 1;
27.
28.  ?>
29.
30.  <!DOCTYPE html>
31.
32.  <html>
33.      <head>
34.          <title>counter</title>
35.      </head>
36.      <body>
37.          You have visited this site <?= $counter ?> time(s).
38.      </body>
39.  </html>
```

```
1. <?php
2.
3.     // configuration
4.     require("../includes/config.php");
5.
6.     // render portfolio
7.     render("form.php");
8.
9. ?>
```

```
1. <?php
2.
3.     require("../includes/config.php");
4.
5.     // ensure user has provided name
6.     if (empty($_POST["name"]))
7.     {
8.         apologize("Missing name!");
9.     }
10.
11.    // ensure user has provided number
12.    if (empty($_POST["number"]))
13.    {
14.        apologize("Missing number!");
15.    }
16.
17.    // ensure user has provided carrier
18.    if (empty($_POST["carrier"]))
19.    {
20.        apologize("Missing carrier!");
21.    }
22.
23.    // remove non-digits from number
24.    $number = preg_replace("/[^\\d]/", "", $_POST["number"]);
25.
26.    // ensure number is 10 digits
27.    if (strlen($number) != 10)
28.    {
29.        apologize("Invalid number!");
30.    }
31.
32.    // INSERT row into users table
33.    query("INSERT INTO users (name, number, carrier) VALUES(?, ?, ?)", $_POST["name"], $number, $_POST["carrier"]);
34.
35.    // report success
36.    render("success.php");
37.
38. ?>
```

```
1. <h1>Error</h1>
2. <p>
3.     <?= htmlspecialchars($message) ?>
4. </p>
```

1. `</body>`
- 2.
3. `</html>`

```
1. <form action="register.php" method="post">
2.     <input name="name" placeholder="Name" type="text"/>
3.     <input name="number" placeholder="Phone Number" type="text"/>
4.     <select name="carrier">
5.         <option value=""></option>
6.         <option value="AT&T">AT&T</option>
7.         <option value="Sprint">Sprint</option>
8.         <option value="T-Mobile">T-Mobile</option>
9.         <option value="Verizon">Verizon</option>
10.    </select>
11.    <input type="submit" value="Register"/>
12. </form>
```

```
1. <!DOCTYPE html>
2.
3. <html>
4.
5.     <head>
6.         <title>lecture</title>
7.     </head>
8.
9.     <body>
```

```
1. <h1>Success</h1>
2. <p>
3.     Added user to database.
4. </p>
```