

CT 310 Midterm 2 – Spring 2008

Name ___ ANSWER KEY _____

EID _____

Question	Max Points	Points
1	15	
2	10	
3	10	
4	10	
5	5	
6	10	
7	10	
8	10	
9	20	
TOTAL	100	

Question 1: Short answer. (15 Points)

Do you agree with the following statement: "Using HTML forms, there is a fundamental limit that a single web page can only direct action, results from submitting information, to a single 'receiving' page." Explain your answer.

This statement is false. It is true that each form defined by a start and end tag has a unique action, but as illustrated by the file browser, one page may contain multiple forms with distinct action pages. *This question is open to two different interpretations, the one intended above, and the other more limited about what happens per submit. Therefore, all answers were scored as correct.*

What does the '@' in '@mysql_select_db' do?

It suppresses error messages.

Explain precisely the security weakness indicated by the following read out from the mysql user table.

```
mysql> SELECT Host, User, Password FROM user;
+-----+-----+-----+
| Host      | User      | Password                                     |
+-----+-----+-----+
| localhost | root      | *D8D923AB65A31D5545120FC293A7D5B12DBEEAB4 |
| %         | root      |                                             |
| localhost | ross      | *5C855820D8E614FD2F0BD901B8711CA3FD2917D4 |
+-----+-----+-----+
6 rows in set (0.02 sec)
```

The second row indicates that there is a host user combination where anyone can log in as 'root' from any host, indicated by the wild card '%', without a password. This leaves the front door wide open.

When a computer running a mysql server is turned off, where does the data reside.

Typically it resides in files on the host server machines file system.

Is MySQL best described as a flat, hierarchical or relational database?

Relational.

Question 2: Slide Show Question – Case Study in Real World Bugs. (10 Points)

There was a bug on the CT310 website over the weekend of April 5th and 6th. Slides were being displayed in an apparently random order. The key bit of PHP code involved in the error was is shown below.

```
function getSlideNames($dir) {
    $i = 0;
    if (is_dir($dir)) {
        if ($dh = opendir($dir)) {
            while (($file = readdir($dh)) !== false) {
                if (strpos($file, "Slide") !== FALSE) {
                    $slides[$i] = $file;
                    $i++;
                }
            }

            closedir($dh);
        }
    }

    /* credit only if solution would run as presented */
    sort($slides);
    return $slides;
}
```

As a bit of background, Systems did a major file server upgrade Friday night, April 4th. Apparently the bug arose after this upgrade.

Part 1: What is your best guess about the cause of the error in slide order (5 Points).

Best guess is that when systems copied all files to a new machine, the new file server, inode order got scrambled in such a way that the directory gave files to PHP in a randomized rather than lexicographic order.

Part 2: Write into the code above a fix for the bug and pay attention to proper syntax, your fix must be precise and correct. The following web snapshot is provided as a memory assist. (5 Points)

Description

```
bool sort ( array &$array [, int $sort_flags ] )
```

This function sorts an array. Elements will be arranged from lowest to highest when this function has completed.

Note: This function assigns new keys for the elements in `array`. It will remove any existing keys you may have assigned, rather than just reordering the keys.

Question 3: Managing State Between the Client and the Server (10 Points)

Part 1: Which version below is a correct header for a PHP generated web page that will maintain session variables. (5 Pages)

Version 1
<pre><?php session_start(); echo '<?xml version="1.0" encoding="utf-8"?>'; echo "\n"; echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">'; ?></pre>
Version 2
<pre><?php echo '<?xml version="1.0" encoding="utf-8"?>'; echo "\n"; echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">'; session_start(); ?></pre>

Correct version is 1.

The reason is **The 'session_start' command must come BEFORE any html is generated.**

Part 2: The CT310 Website displays the IP address of the client that has established a session. This is displayed in the lower left corner of each page. The code to generate this display is:

```
Originating IP <?php echo $_SESSION['ipaddress'] ?>
```

Does the server store this session information in cookies downloaded to the client?

Backup your answer with a concise description of the actual process involved in maintaining state information while a user on a client browser moves between pages on the CT310 website. (5 Points)

NO.

As demonstrated in lecture, there is only one cookie and this has a unique session identifier. The session variables associated with that unique session identifier are stored on the server. An advanced note, it is possible to configure PHP to not even use this cookie arrangement, but that was not demonstrated.

Question 4: PHP Site Layout (10 Points)

Part 1: About two weeks ago, a new link 'Colvin' was added to the banner on all the CT310 web pages reached through www.cs.colostate.edu/~ct310. How many files required modification to make this change? (5 Points)

One, the file `ct310topbanner.php`.

Part 2: The following is a code snippet extracted from the page `ct310topbanner.php`

```
switch ($pgLevel) {  
    case 1: $relPath = "../";    break;  
    case 2: $relPath = "../../"; break;  
    case 3: $relPath = "../../../"; break;  
    default: $relPath = "../";  
}
```

What is the purpose of this code? (5 Points)

The PHP variable `$pgLevel` is set by the page including the banner, and then this code is used to generate an appropriate relative path, the variable `$relPath`, to other pages on the ct310 web site. The purpose is that this mechanism allows a single banner PHP file to be included at different levels of the site hierarchy and behave as needed to make relative links always work.

Question 5: PHP date and time. (5 Points)

The `mktime` function plays a critical role in the automated creation of the calendar on the CT310 progress page. Here is a simple bit of PHP that takes this function out for a spin.

```
$td = date("F j", mktime(0, 0, 0, 2, 329, 2008));  
echo "<p>The 329th day in February 2008 is: $td</p> \n";
```

Which of the following happens with this PHP code is run:

1. There is an error, since February 2008 has only 29 days.
2. The page displays: The 329th day in February 2008 is: February 29
3. The page displays: The 329th day in February 2008 is: December 25
4. The page displays: The 329th day in February 2008 is:

The answer is 3, 'December 25'

You might well ask, how am I supposed to know that? The answer is that in reviewing for the midterm attention was drawn to the page `Progress.php`. The specific code snippet on the Slide Packet 11, Slide 3, shows where `mktime` is used to generate the current date. Above this point in the PHP code is code to generate the start and end dates of each week. That code takes explicit advantage of the somewhat unusual behavior of PHP to increment months when specifying days, say day 329, of a month. This a major feature, since absent this behavior, automated generation of dates such as a string of Sundays, would be very difficult.

Question 6: File Browser Vulnerability (10 Points)

There was a weakness in the first version of the file browser and download facility added to the CT310 website. The vulnerability involved this following code:

File zdownDoit.php
<pre>if (isset(\$_POST['file'])) { \$file = \$_POST['file']; header("Content-type: application/force-download"); header("Content-Transfer-Encoding: Binary"); header("Content-length: ".filesize(\$file)); header("Content-disposition: attachment; filename=\"\".basename(\$file).\""); readfile("\$file"); }</pre>

Part 1: Describe in a few brief sentences the exploit, and in particular how it could be used to download a world readable file anywhere in the CS file system. (5 Points)

First, anyone in the world can write a webpage with a form that passes information to this page in the variable 'file'. Next, since the code does no checking of the contents of 'file', were someone outside to know what they were looking for, they might easily guess a path that would then be downloadable. For example, many Unix installations store user passwords at /etc/passwd.

Part 2: Describe in a few brief sentences the fix presented in class and added to the file zdownDoIt.php. You need NOT write code to answer this question, just be succinct in how to correct the vulnerability. (5 Points)

There are two checks added to make the download more secure. First, insist that the path begin with the site root written as './'. Second, do NOT allow '..' to appear anywhere in the path. The full fixed code appears on the next page.

Question 6, Part 2, Answer Continued

Fixed File zdownDoit.php

```
<?php

/* File name has to pass a series of tests before it can be safely downloaded.

   First, the file name must begin with './' otherwise it may be an absolute
   address and enable someone to grab files outside the site tree.

   Second, regardless of how it starts, it cannot contain an up directive '..'.
   */

function fileSafe ($file) {
    $pos = strpos($file, './');
    if ($pos == 0 && (! ($pos === false))) {
        $startOK = true; }
    else {
        $startOK = false; }

    $pos = strpos($file, '..');
    if ($pos === false) {
        $upFree = true; }
    else {
        $upFree = false; }

    if ($startOK && $upFree) {
        $safe = true; }
    else {
        $safe = false; }

    return $safe;
}

if (isset($_POST['file'])) {
    $file = $_POST['file'];
    if (fileSafe($file)) {
        header("Content-type: application/force-download");
        header("Content-Transfer-Encoding: Binary");
        header("Content-length: ".filesize($file));
        header("Content-disposition: attachment; filename=\"\".basename($file).\"\"");
        readfile("$file");
    }
    else {
        echo "Download not permitted for file $file \n";
    }
}

?>
```

Question 7: Basic MySQL (10 Points)

Consider the following MySQL database setup.

```
CREATE TABLE kind (
  kind_id int(5) NOT NULL,
  kind_name varchar(50),
  PRIMARY KEY (kind_id) );
CREATE TABLE breed (
  breed_id int(5) NOT NULL,
  breed_name varchar(50),
  PRIMARY KEY (breed_id) );
CREATE TABLE pname (
  pname_id int(5) NOT NULL,
  callme varchar(50),
  PRIMARY KEY (pname_id) );
CREATE TABLE pets (
  pet_id int(5) NOT NULL AUTO_INCREMENT,
  kind_id int(5) NOT NULL,
  breed_id int(5) NOT NULL,
  pname_id int(5) NOT NULL,
  PRIMARY KEY (pet_id),
  KEY breed (breed_id),
  KEY kind (kind_id),
  KEY pname (pname_id) );
INSERT INTO kind VALUES
(1, 'Bird'),
(2, 'Cat'),
(3, 'Dog');
INSERT INTO breed VALUES
(1, 'Beagle'),
(2, 'Persian'),
(3, 'Hornbill');
INSERT INTO pname VALUES
(1, 'Snoopy'),
(2, 'Tiger'),
(3, 'Zazu');
```

Complete the following command so that the SELECT shown on the next page works.

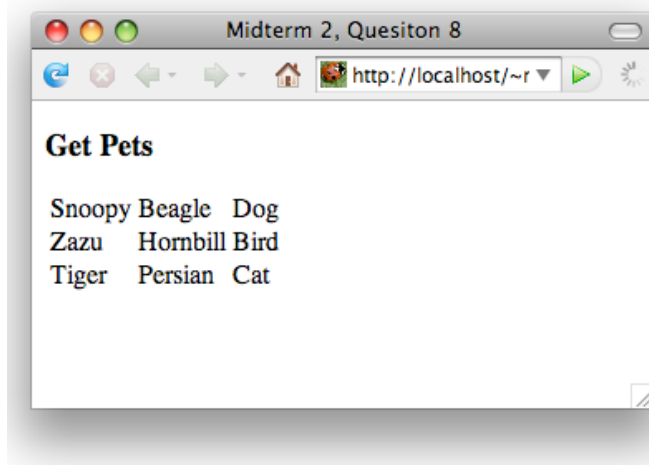
```
INSERT INTO pets (kind_id, breed_id, pname_id) VALUES
(3, 1, 1),
(2, 2, 2),
(1, 3, 3);
```

Question 7 Continued.

```
mysql> SELECT callme, breed_name, kind_name FROM pets
      -> NATURAL JOIN pname NATURAL JOIN breed NATURAL JOIN kind;
+-----+-----+-----+
| callme | breed_name | kind_name |
+-----+-----+-----+
| Snoopy | Beagle     | Dog       |
| Tiger  | Persian    | Cat       |
| Zazu   | Hornbill   | Bird      |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Question 8: PHP Talks to MySQL (10 Points)

Consider the database created in Question 7. Here is a web page and the associated PHP code with a portion missing. Fill in the missing code.



```
<h3>Get Pets</h3>
```

```
<?php
```

```
$conn = mysql_connect('localhost', 'ct310', 'mydog');  
@mysql_select_db('pets', $conn) or die('Cannot relate to pets.');
```

```
$query = "SELECT callme, breed_name, kind_name FROM pets".  
        " NATURAL JOIN pname NATURAL JOIN breed NATURAL JOIN kind".  
        " ORDER BY breed_name";  
$result = mysql_query($query);  
mysql_close($conn);  
echo '<table>'. "\n";  
for ($i = 0; $i < mysql_numrows($result); $i++) {  
    $name = mysql_result($result, $i, 'callme');  
    $breed = mysql_result($result, $i, 'breed_name');  
    $kind = mysql_result($result, $i, 'kind_name');  
    echo "<tr><td>$name</td><td>$breed</td><td>$kind</td></tr>\n";
```

```
    }  
    echo '</table>'. "\n";
```

```
?>  
</body>  
</html>
```

Question 9: Wakeup and smell the ... (20 Points)

Here is a MySQL script that establishes a database for a toy online coffee distributor.

```
use happy_buzz;

DROP TABLE IF EXISTS brands;
DROP TABLE IF EXISTS inventory;

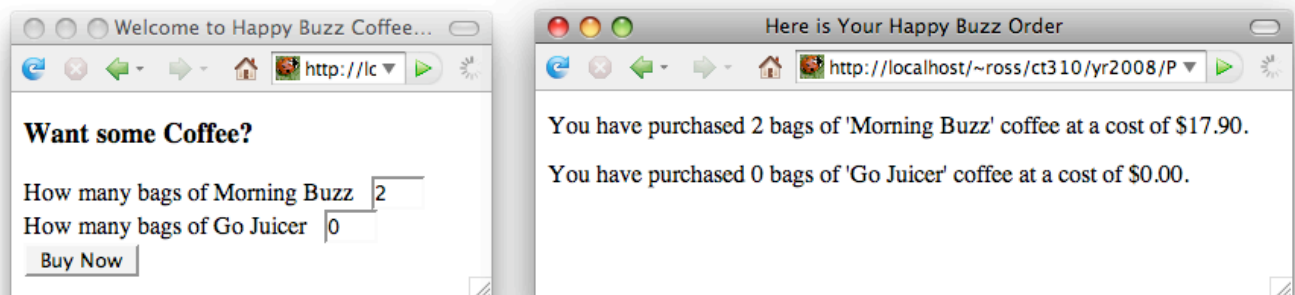
CREATE TABLE brands (
  brand_id    int(5) NOT NULL,
  brand_name  varchar(50),
  price_cents int(10) NOT NULL,
  PRIMARY KEY (brand_id) );

CREATE TABLE inventory (
  track      int(5) NOT NULL AUTO_INCREMENT,
  brand_id   int(5) NOT NULL,
  instock    int(10) NOT NULL,
  KEY brands (brand_id),
  PRIMARY KEY (track) );

INSERT INTO brands VALUES
  (1, 'Morning Buzz', 895),
  (2, 'Go Juicer', 1295);

INSERT INTO inventory (brand_id, instock) VALUES
  (1, 100),
  (2, 1000);
```

Here is an example of two web pages than allow one to 'purchase' coffee.



The following page presents the PHP source code for these pages, with some choice bits left out, and asks you three questions.

Question 9 Continued

Part 1: Here is the file `coffeeSelect.php` with some of the needed code missing. Fill in the missing parts (10 Points).

```
11 </head>
12 <body>
13 <h3>Want some Coffee?</h3>
14 <?php
15     $conn = mysql_connect('localhost', 'ct310', 'mydog');
16     @mysql_select_db(
17         $query = "SELECT brand_name AS name FROM brands";
18         $result = mysql_query($query);
19         mysql_close($conn);
20     ?>
21     <form
22         How many bags of <?php echo mysql_result($result, 0, 'name')?> &nbsp;
23
24
25
26
27     </form>
28 </body>
29 </html>
```

Here is the code with the missing bits filled in ...

```
11 </head>
12 <body>
13 <h3>Want some Coffee?</h3>
14 <?php
15     $conn = mysql_connect('localhost', 'ct310', 'mydog');
16     @mysql_select_db('happy_buzz', $conn) or die('Cannot connect to happy_buzz.');
```

```
17     $query = "SELECT brand_name AS name FROM brands";
18     $result = mysql_query($query);
19     mysql_close($conn);
20 ?>
21 <form action="coffeeBuy" method="post">
22     How many bags of <?php echo mysql_result($result, 0, 'name')?> &nbsp;
23     <input type="text" size="3" value="0" name="bags_brand_1"> <br/>
24     How many bags of <?php echo mysql_result($result, 1, 'name')?> &nbsp;
25     <input type="text" size="3" value="0" name="bags_brand_2"> <br/>
26     <input type="submit" value="Buy Now">
27 </form>
28 </body>
29 </html>
```

Question 9 Continued

Part 2: Here is the code for the file coffeeBuy.php. This code is complete, but does not actually modify the inventory to reflect the purchase. Write down below in the box provided precisely the PHP code required to correct this omission. (10 Points)

```
12 <body>
13 <?php
14     IF (isset($_POST['bags_brand_1'])) $bags[1] = $_POST['bags_brand_1'];
15     IF (isset($_POST['bags_brand_2'])) $bags[2] = $_POST['bags_brand_2'];
16
17     $conn = mysql_connect('localhost', 'ct310', 'mydog');
18     @mysql_select_db('happy_buzz', $conn) or die('Cannot connect to happy_buzz.');
```

19 for (\$i = 1; \$i < 3; \$i++) {

```
20         $query = "SELECT brand_name, price_cents, instock FROM inventory NATURAL JOIN brands".
21                 " WHERE brand_id = ".$i;
22         $result = mysql_query($query);
23         $brand_name[$i] = mysql_result($result, 0, 'brand_name');
24         $brand_cents[$i] = mysql_result($result, 0, 'price_cents');
25         $brand_stock[$i] = mysql_result($result, 0, 'instock');
26         $cost[$i] = sprintf("%01.2f", ($brand_cents[$i] * $bags[$i]) / 100.0);
27     }
28     mysql_close($conn);
29 ?>
30     <p>You have purchased <?php echo $bags[1]?> bags of
31     '<?php echo $brand_name[1]?>' coffee at a cost of
32     $<?php echo $cost[1]?>.</p>
33     <p>You have purchased <?php echo $bags[2]?> bags of
34     '<?php echo $brand_name[2]?>' coffee at a cost of
35     $<?php echo $cost[2]?>.</p>
36 </body>
37 </html>
```

New PHP code to correct inventory

```
<?php
    $conn = mysql_connect('localhost', 'ct310', 'ct310');
    @mysql_select_db('happy_buzz', $conn)
        or die('Cannot connect to happy_buzz.');
```

for (\$i = 1; \$i < 3; \$i++) {

```
    $new_instock = $brand_stock[$i] - $bags[$i];
    $query = "UPDATE inventory SET instock=".$new_instock.
            " WHERE brand_id = ".$i;
    mysql_query($query);
}
```

```
mysql_close($conn);
?>
```