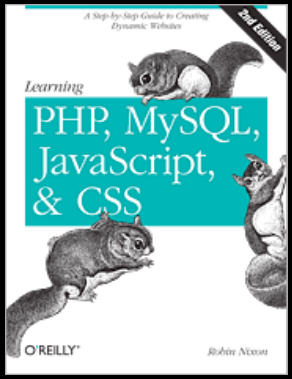
Accessing MySQL Using PDO

Chapter 10

Dr. Charles Severance

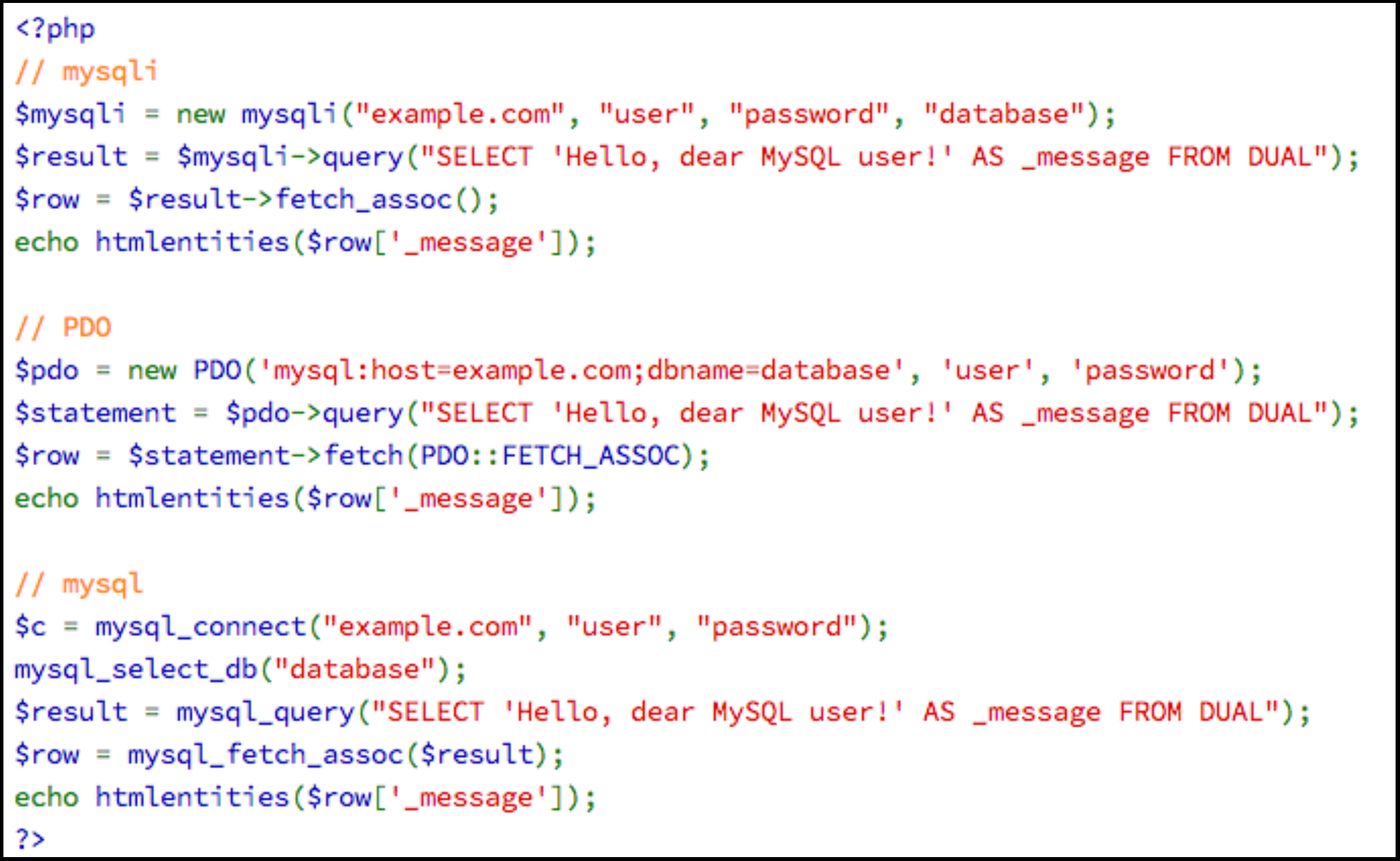

A Step-by-Step Guide to Creating Dynamic Websites
2nd Edition
Learning
PHP, MySQL, JavaScript, & CSS
O'REILLY
Robin Nixon

To be used in association with the book:
PHP, MySql, and JavaScript by Robin Nixon

Multiple Ways to Access MySql

- PHP is evolving - there are three ways to access MySql from PHP
  - Legacy non-OO mysql_ routines (deprecated)
  - New mysqli (OO version that is similar to mysql_)
  - PDO - Portable Data Objects
- A perfect topic for debate

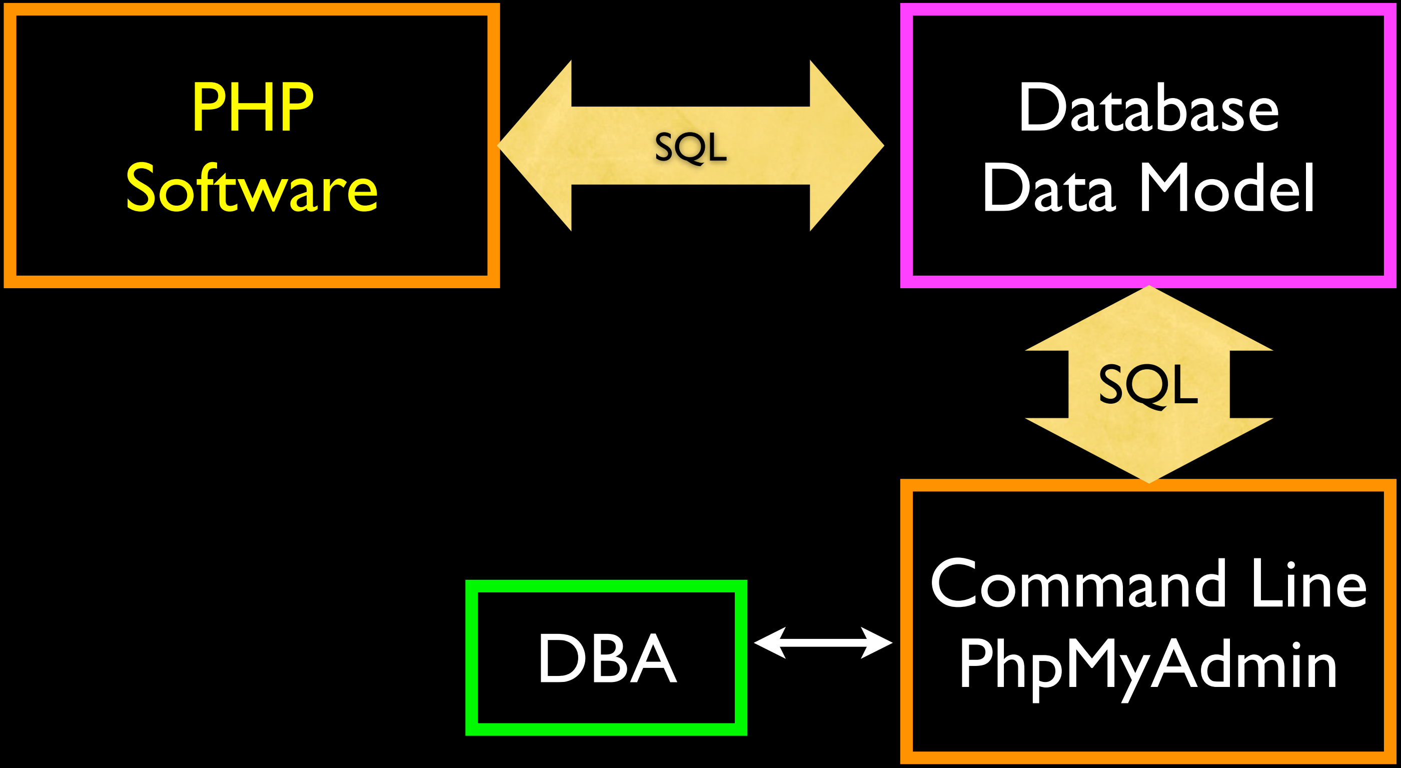
http://php.net/manual/en/mysqlinfo.api.choosing.php



```php
<?php
// mysqli
$mysqli = new mysqli("example.com", "user", "password", "database");
$result = $mysqli->query("SELECT 'Hello, dear MySQL user!' AS _message FROM DUAL");
$row = $result->fetch_assoc();
echo htmlentities($row['_message']);

// PDO
$pdo = new PDO('mysql:host=example.com;dbname=database', 'user', 'password');
$statement = $pdo->query("SELECT 'Hello, dear MySQL user!' AS _message FROM DUAL");
$row = $statement->fetch(PDO::FETCH_ASSOC);
echo htmlentities($row['_message']);

// mysql
$c = mysql_connect("example.com", "user", "password");
mysql_select_db("database");
$result = mysql_query("SELECT 'Hello, dear MySQL user!' AS _message FROM DUAL");
$row = mysql_fetch_assoc($result);
echo htmlentities($row['_message']);
?>
```

http://php.net/manual/en/mysqlinfo.api.choosing.php

Application Structure



Creating a Database and User

- CREATE DATABASE misc;
- GRANT ALL ON misc.* TO 'fred'@'localhost' IDENTIFIED BY 'zap';
- GRANT ALL ON misc.* TO 'fred'@'127.0.0.1' IDENTIFIED BY 'zap';
- USE misc;   (if you are in the command line)

```
/Applications/MAMP/Library/bin/mysql -u root -P 8889 -p
/Applications/xampp/xamppfiles/bin/mysql -u root -p
```

```
c:\xampp\mysql\bin\mysql.exe
```

Creating a Table

```
CREATE TABLE users (
  id INT UNSIGNED NOT NULL
     AUTO_INCREMENT KEY,
  name VARCHAR(128),
  email VARCHAR(128),
  password VARCHAR(128));

ALTER TABLE users ADD INDEX(email);
```

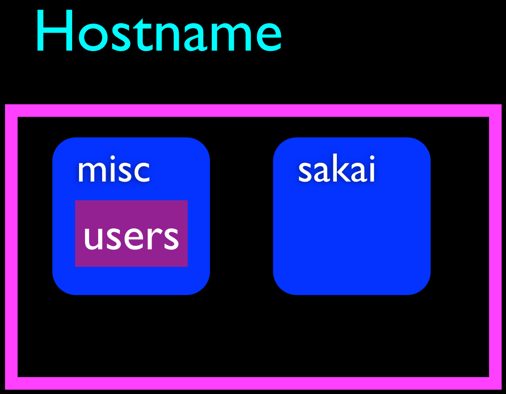
```
mysql> describe users;
+----------+------------------+------+-----+---------+----------------+
| Field    | Type             | Null | Key | Default | Extra          |
+----------+------------------+------+-----+---------+----------------+
| id       | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| name     | varchar(128)     | YES  |     | NULL    |                |
| email    | varchar(128)     | YES  | MUL | NULL    |                |
| password | varchar(128)     | YES  |     | NULL    |                |
+----------+------------------+------+-----+---------+----------------+
```

Inserting a Few Records

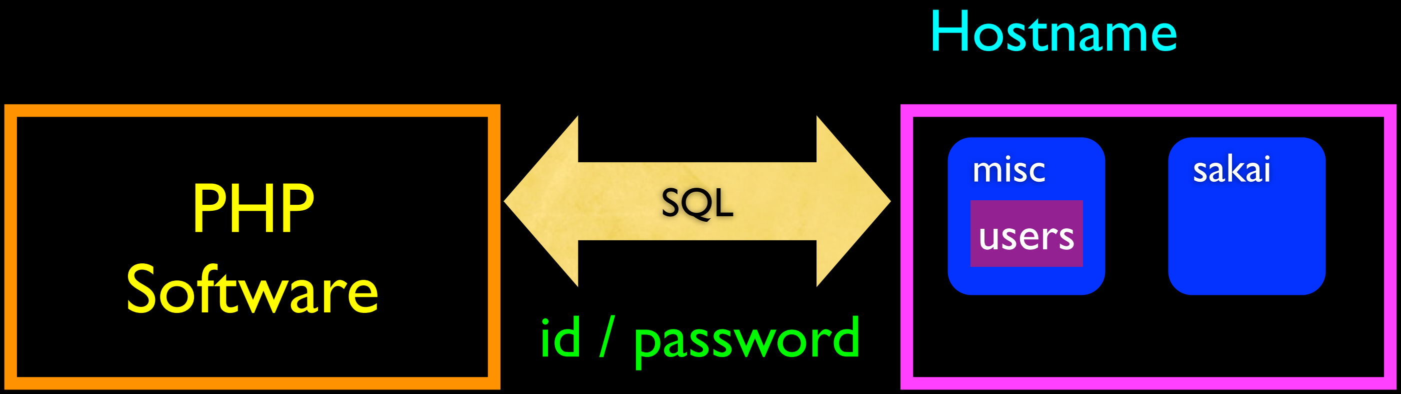
```
INSERT INTO users (name,email,password) VALUES ('Chuck','csev@umich.edu','123');
INSERT INTO users (name,email,password) VALUES ('Glenn','gg@umich.edu','456');
```

```
mysql> select * from users;
+----+-------+----------------+----------+
| id | name  | email          | password |
+----+-------+----------------+----------+
|  1 | Chuck | csev@umich.edu | 123      |
|  2 | Glenn | gg@umich.edu   | 456      |
+----+-------+----------------+----------+
```

Database Connection



Database Connection



```
$db = new PDO('mysql:host=localhost;port=8889;dbname=misc',
    'fred', 'zap');
```

db.php

first.php

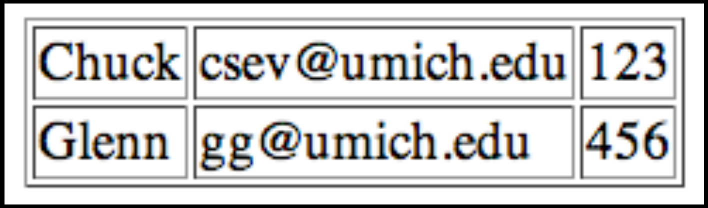
```
<?php
echo "<pre>\n";
$pdo=new PDO('mysql:host=localhost;port=8889;dbname=misc',
    'fred', 'zap');
$stmt = $pdo->query("SELECT * FROM users");
while ( $row = $stmt->fetch(PDO::FETCH_ASSOC) ) {
    print_r($row);
}
echo "</pre>\n";
?>
```

```
mysql> select * from users;
+----+-------+-----------------+----------+
| id | name  | email           | password |
+----+-------+-----------------+----------+
|  1 | Chuck | csev@umich.edu  | 123      |
|  2 | Glenn | gg@umich.edu    | 456      |
+----+-------+-----------------+----------+
```

```
Array
(
    [id] => 1
    [name] => Chuck
    [email] => csev@umich.edu
    [password] => 123
)
Array
(
    [id] => 2
    [name] => Glenn
    [email] => gg@umich.edu
    [password] => 456
)
```

second.php

```php
<?php
$pdo = new PDO('mysql:host=localhost;port=8889;dbname=misc',
    'fred', 'zap');
$stmt = $pdo->query("SELECT name, email, password FROM users");
echo '<table border="1">'."\n";
while ( $row = $stmt->fetch(PDO::FETCH_ASSOC) ) {
    echo "<tr><td>";
    echo($row['name']);
    echo("</td><td>");
    echo($row['email']);
    echo("</td><td>");
    echo($row['password']);
    echo("</td></tr>\n");
}
echo "</table>\n";
?>
```



Chuck	csev@umich.edu	123
Glenn	gg@umich.edu	456

```html
<table border="1">
<tr><td>Chuck</td><td>csev@umich.edu</td><td>123</td></tr>
<tr><td>Glenn</td><td>gg@umich.edu</td><td>456</td></tr>
</table>
```

Pattern

- Put database connection information in a single file and include it in all your other files
  - Helps make sure to not to mistakenly reveal id / pw
  - Don't check it into a public source repository :)

db.php

```
<?php
$pdo = new PDO('mysql:host=localhost;port=8889;dbname=misc',
   'fred', 'zap');
?>
```

third.php

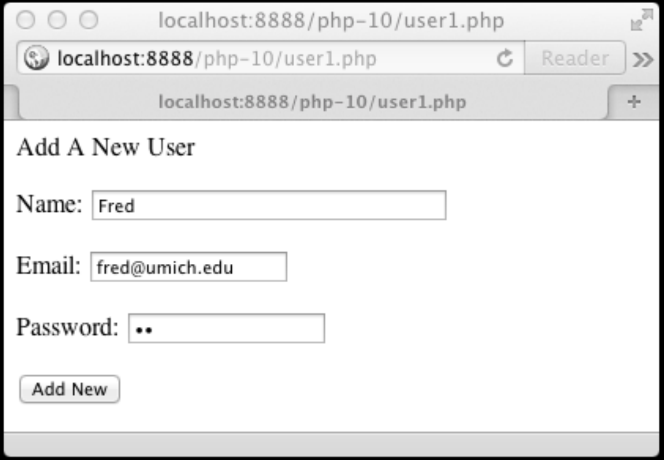
```
<?php
echo "<pre>\n";
require_once "db.php";

$stmt = $pdo->query("SELECT * FROM users");
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)){
    print_r($row);
}
echo "</pre>\n";
?>
```

```
Array
(
    [id] => 1
    [name] => Chuck
    [email] => csev@umich.edu
    [password] => 123
)
Array
(
    [id] => 2
    [name] => Glenn
    [email] => gg@umich.edu
    [password] => 456
)
```

Lets put some data in a database!

user1.php

```
<?php
require_once "db.php";

if ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password'])) {
    $sql = "INSERT INTO users (name, email, password)
              VALUES (:name, :email, :password)";
    echo("<pre>\n".$sql."\n</pre>\n");
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password']));
}
?>
<html>
<head></head><body>
<p>Add A New User</p>
<form method="post">
<p>Name:
<input type="text" name="name" size="40"></p>
<p>Email:
<input type="text" name="email"></p>
<p>Password:
<input type="password" name="password"></p>
<p><input type="submit" value="Add New"/></p>
</form>
</body>
```



localhost:8888/php-10/user1.php

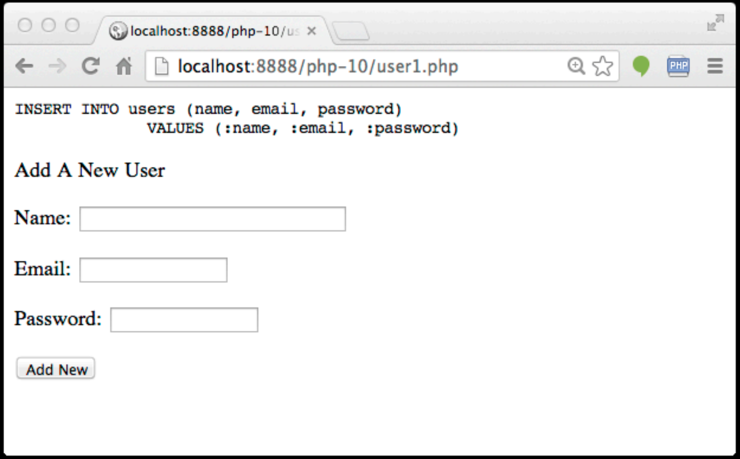
Add A New User

Name: Fred

Email: fred@umich.edu

Password: ••

Add New



localhost:8888/php-10/user1.php

```
INSERT INTO users (name, email, password)
              VALUES (:name, :email, :password)
```

Add A New User

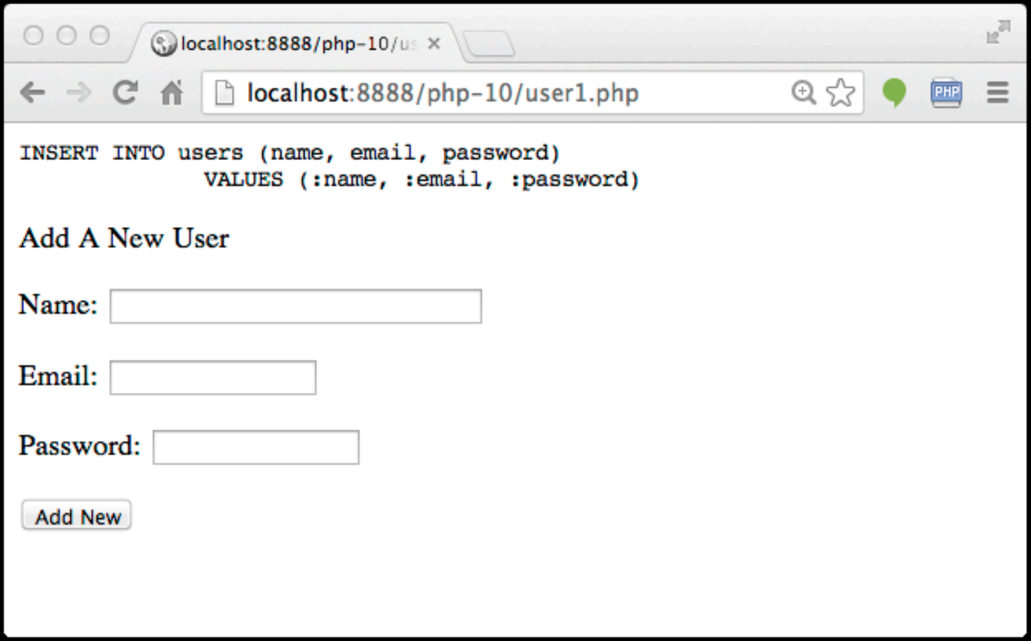
Name:

Email:

Password:

Add New

user1.php



```
INSERT INTO users (name, email, password)
              VALUES (:name, :email, :password)
```

Add A New User

Name:

Email:

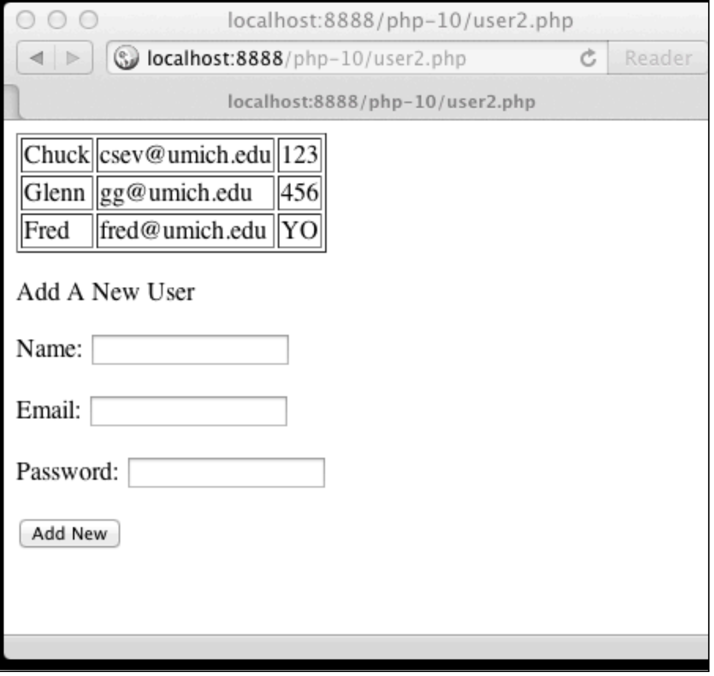
Password:

Add New

```
mysql> select * from users;
+----+-------+----------------+----------+
| id | name  | email          | password |
+----+-------+----------------+----------+
|  1 | Chuck | csev@umich.edu | 123      |
|  2 | Glenn | gg@umich.edu   | 456      |
|  3 | Fred  | fred@umich.edu | YO       |
+----+-------+----------------+----------+
```

user2.php

```
    $sql = "INSERT INTO users (name, email, password)
               VALUES ('$n', '$e', '$p')";
    echo "<pre>\n$sql\n</pre>\n";
    mysql_query($sql);
 }
 <html>
 <head></head><body><table border="1">
 <?php
 $stmt = $pdo->query("SELECT name, email, password
 FROM users");
 while ( $row = $stmt->fetch(PDO::FETCH_ASSOC) ) {
     echo "<tr><td>";
     echo($row['name']);
     echo("</td><td>");
     echo($row['email']);
     echo("</td><td>");
     echo($row['password']);
     echo("</td></tr>\n");
 }
 ?>
 </table>
 <p>Add A New User</p>
```


localhost:8888/php-10/user2.php

Chuck	csev@umich.edu	123
Glenn	gg@umich.edu	456
Fred	fred@umich.edu	YO

Add A New User

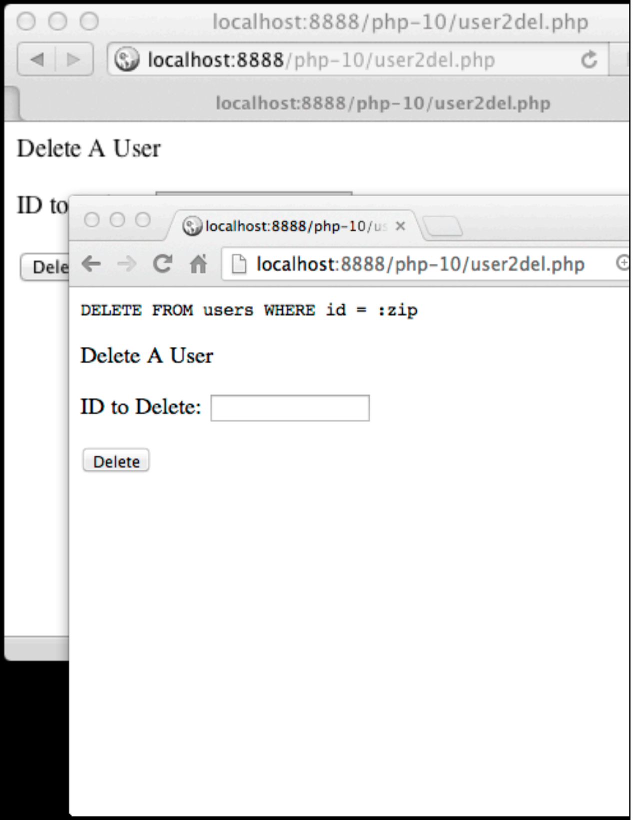
Name:

Email:

Password:

user2del.php

```
<?php
require_once "db.php";

if ( isset($_POST['id']) ) {
    $sql="DELETE FROM users WHERE id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip'=>$_POST['id']));
}
?>
<p>Delete A User</p>
<form method="post">
<p>ID to Delete:
<input type="text" name="id"></p>
<p><input type="submit" value="Delete"/></p>
</form>
```


localhost:8888/php-10/user2del.php
localhost:8888/php-10/user2del.php
localhost:8888/php-10/user2del.php
Delete A User
ID to
Dele
localhost:8888/php-10/u
localhost:8888/php-10/user2del.php

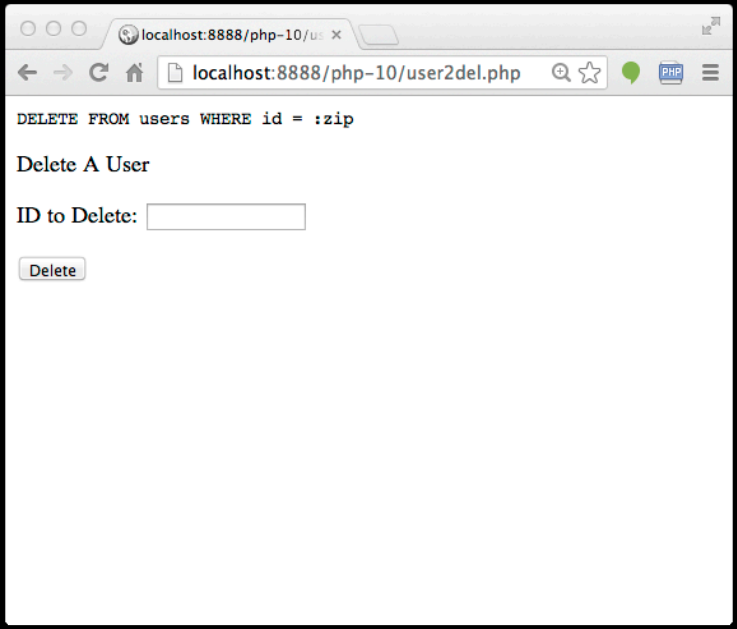
```
DELETE FROM users WHERE id = :zip
```

Delete A User

ID to Delete:

Delete

Don't change data in a GET



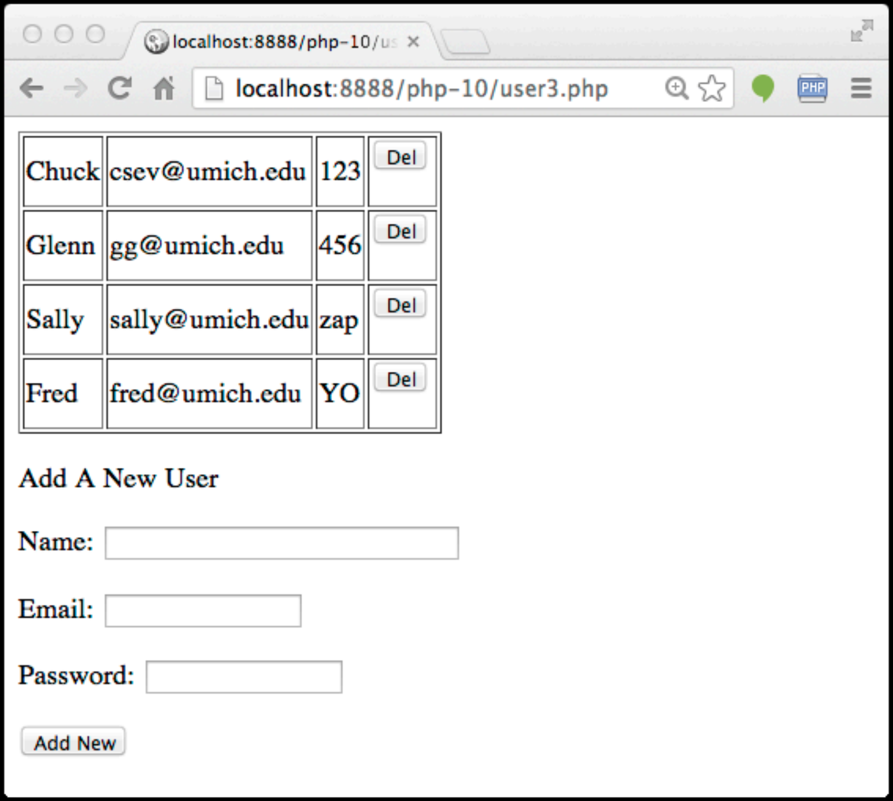
localhost:8888/php-10/user2del.php

```
DELETE FROM users WHERE id = :zip
```

Delete A User

ID to Delete:

```
mysql> select * from users;
+----+-------+----------------+----------+
| id | name  | email          | password |
+----+-------+----------------+----------+
|  1 | Chuck | csev@umich.edu | 123      |
|  2 | Glenn | gg@umich.edu   | 456      |
+----+-------+----------------+----------+
```



Chuck	csev@umich.edu	123	Del
Glenn	gg@umich.edu	456	Del
Sally	sally@umich.edu	zap	Del
Fred	fred@umich.edu	YO	Del

Add A New User

Name:

Email:

Password:

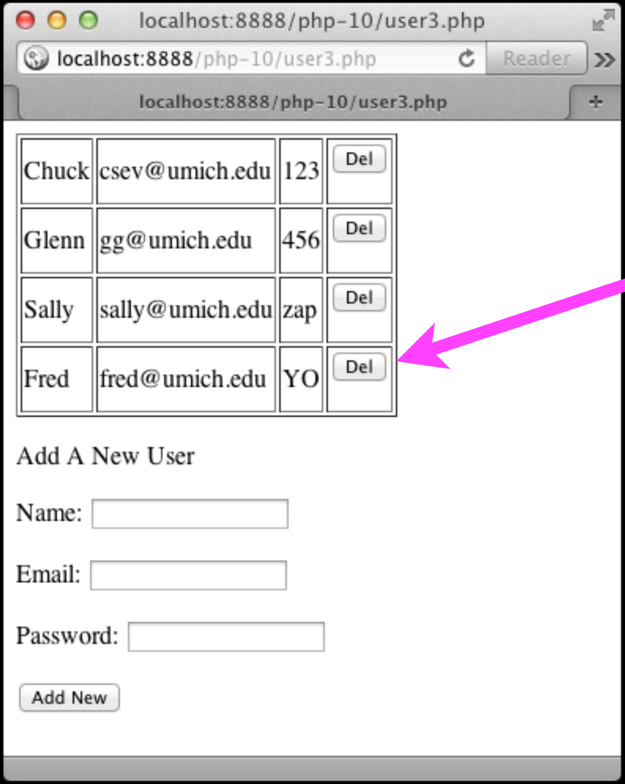
Add New

user3.php

user3.php

```
if ( isset($_POST['delete']) && isset($_POST['id']) ) {
    $sql = "DELETE FROM users WHERE id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['id']));
}
?>
<html>
<head></head><body><table border="1">
<?php
$stmt = $pdo->query("SELECT name, email, password, id FROM users");
while ( $row = $stmt->fetch(PDO::FETCH_ASSOC) ) {
    echo "<tr><td>";
    echo($row['name']);
    echo("</td><td>");
    echo($row['email']);
    echo("</td><td>");
    echo($row['password']);
    echo("</td><td>");
    echo('<form method="post"><input type="hidden" ');
    echo('name="id" value="'.$row['id'].'">'."\n");
    echo('<input type="submit" value="Del" name="delete">');
    echo("\n</form>\n");
    echo("</td></tr>\n");
}
```

```
echo('<form method="post"><input type="hidden" ');
echo('name="id" value="'.$row['id'].'">'."\n");
echo('<input type="submit" value="Del" name="delete">');
echo("\n</form>\n");
```



Chuck	csev@umich.edu	123	Del
Glenn	gg@umich.edu	456	Del
Sally	sally@umich.edu	zap	Del
Fred	fred@umich.edu	YO	Del

Add A New User

Name:

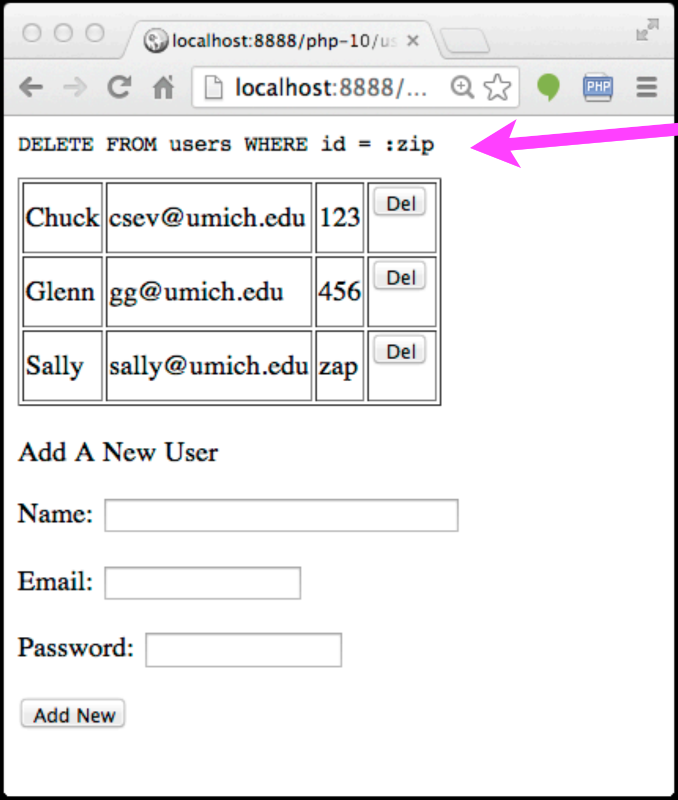
Email:

Password:

Add New

```
<tr><td>Fred</td><td>fred@umich.edu</td>
<td>YO</td>
<td>
<form method="post">
<input type="hidden" name="id" value="5">
<input type="submit" value="Del" name="delete">
</form>
</td>
</tr>
```

```
if ( isset($_POST['delete']) && isset($_POST['id']) )
    $sql = "DELETE FROM users WHERE id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['id']));
}
```



```
DELETE FROM users WHERE id = :zip
```

Chuck	csev@umich.edu	123	Del
Glenn	gg@umich.edu	456	Del
Sally	sally@umich.edu	zap	Del

Add A New User

Name:

Email:

Password:

Add New

```
if ( isset($_POST['delete']) && isset($_POST['id']) ) {
    $sql = "DELETE FROM users WHERE id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['id']));
}
```

Program Outline

```
<?php
require_once "db.php";

if ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password'])) {
    $sql = "INSERT INTO users (name, email, password)
              VALUES (:name, :email, :password)";
    echo("<pre>\n".$sql."\n</pre>\n");
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password']));
}

if ( isset($_POST['delete']) && isset($_POST['id']) ) {
    $sql = "DELETE FROM users WHERE id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['id']));
}
?>
```

```
<?php
require_once "db.php";

if ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password'])) {
    $sql = "INSERT INTO users (name, email, password)
              VALUES (:name, :email, :password)";
    echo("<pre>\n".$sql."\n</pre>\n");
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password']));
}

if ( isset($_POST['delete']) && isset($_POST['id']) ) {
    $sql = "DELETE FROM users WHERE id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['id']));
}
?>
<html>
<head></head><body><table border="1">
<?php
$stmt = $pdo->query("SELECT name, email, password, id FROM users");
while ( $row = $stmt->fetch(PDO::FETCH_ASSOC) ) {
    echo "<tr><td>";
    echo($row['name']);
    echo("</td><td>");
    echo($row['email']);
    echo("</td><td>");
    echo($row['password']);
    echo("</td><td>");
    echo('<form method="post"><input type="hidden" ');
    echo('name="id" value="'.$row['id'].'">'."\n");
    echo('<input type="submit" value="Del" name="delete">');
    echo("\n</form>\n");
    echo("</td></tr>\n");
}
?>
</table>
<p>Add A New User</p>
<form method="post">
<p>Name:
<input type="text" name="name" size="40"></p>
<p>Email:
<input type="text" name="email"></p>
<p>Password:
<input type="password" name="password"></p>
<p><input type="submit" value="Add New"/></p>
</form>
</body>
```

```
<html>
<head></head><body><table border="1">
<?php
$stmt = $pdo->query("SELECT name, email, password, id FROM users");
while ( $row = $stmt->fetch(PDO::FETCH_ASSOC) ) {
    echo "<tr><td>";
    echo($row['name']);
    echo("</td><td>");
    echo($row['email']);
    echo("</td><td>");
    echo($row['password']);
    echo("</td><td>");
    echo('<form method="post"><input type="hidden" ');
    echo('name="id" value="'.$row['id'].'">'."\n");
    echo('<input type="submit" value="Del" name="delete">');
    echo("\n</form>\n");
    echo("</td></tr>\n");
}
?>
</table>
```

```
<?php
require_once "db.php";

if ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password'])) {
    $sql = "INSERT INTO users (name, email, password)
              VALUES (:name, :email, :password)";
    echo("<pre>\n".$sql."\n</pre>\n");
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password']));
}

if ( isset($_POST['delete']) && isset($_POST['id']) ) {
    $sql = "DELETE FROM users WHERE id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['id']));
}
?>
<html>
<head></head><body><table border="1">
<?php
$stmt = $pdo->query("SELECT name, email, password, id FROM users");
while ( $row = $stmt->fetch(PDO::FETCH_ASSOC) ) {
    echo "<tr><td>";
    echo($row['name']);
    echo("</td><td>");
    echo($row['email']);
    echo("</td><td>");
    echo($row['password']);
    echo("</td><td>");
    echo('<form method="post"><input type="hidden" ');
    echo('name="id" value="'.$row['id'].'">'."\n");
    echo('<input type="submit" value="Del" name="delete">');
    echo("\n</form>\n");
    echo("</td></tr>\n");
}
?>
</table>
<p>Add A New User</p>
<form method="post">
<p>Name:
<input type="text" name="name" size="40"></p>
<p>Email:
<input type="text" name="email"></p>
<p>Password:
<input type="password" name="password"></p>
<p><input type="submit" value="Add New"/></p>
</form>
</body>
```

Program Outline

```
<p>Add A New User</p>
<form method="post">
<p>Name:
<input type="text" name="name" size="40"></p>
<p>Email:
<input type="text" name="email"></p>
<p>Password:
<input type="password" name="password"></p>
<p><input type="submit" value="Add New"/></p>
</form>
</body>
```

```
<?php
require_once "db.php";

if ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password'])) {
    $sql = "INSERT INTO users (name, email, password)
              VALUES (:name, :email, :password)";
    echo("<pre>\n".$sql."\n</pre>\n");
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password']));
}

if ( isset($_POST['delete']) && isset($_POST['id']) ) {
    $sql = "DELETE FROM users WHERE id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['id']));
}
?>

<html>
<head></head><body><table border="1">
<?php
$stmt = $pdo->query("SELECT name, email, password, id FROM users");
while ( $row = $stmt->fetch(PDO::FETCH_ASSOC) ) {
    echo "<tr><td>";
    echo($row['name']);
    echo("</td><td>");
    echo($row['email']);
    echo("</td><td>");
    echo($row['password']);
    echo("</td><td>");
    echo('<form method="post"><input type="hidden" ');
    echo('name="id" value="'.$row['id'].'">'."\n");
    echo('<input type="submit" value="Del" name="delete">');
    echo("\n</form>\n");
    echo("</td></tr>\n");
}
?>
</table>
<p>Add A New User</p>
<form method="post">
<p>Name:
<input type="text" name="name" size="40"></p>
<p>Email:
<input type="text" name="email"></p>
<p>Password:
<input type="password" name="password"></p>
<p><input type="submit" value="Add New"/></p>
</form>
</body>
```

Security Alert: SQL Injection

SQL injection or SQLi is a code injection technique that exploits a security vulnerability in some computer software. An injection occurs at the database level of an application (like queries). The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. Using well designed query language interpreters can prevent SQL injections.

http://en.wikipedia.org/wiki/SQL_injection

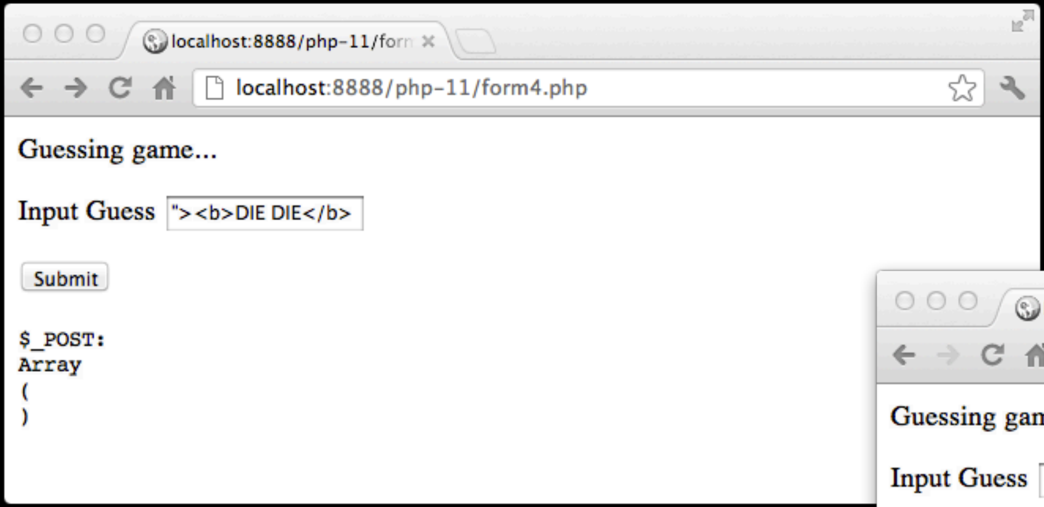
Security Alert: SQL Injection

- This (deprecated) code is prone to SQL Injection - where?

```
$e = $_POST['email'];
$p = $_POST['password'];
$sql = "SELECT name FROM users
    WHERE email = '$e'
    AND password = '$p'";

$result = mysql_query($sql);
$row = mysql_fetch_row($result);
```

login1.php

Recall



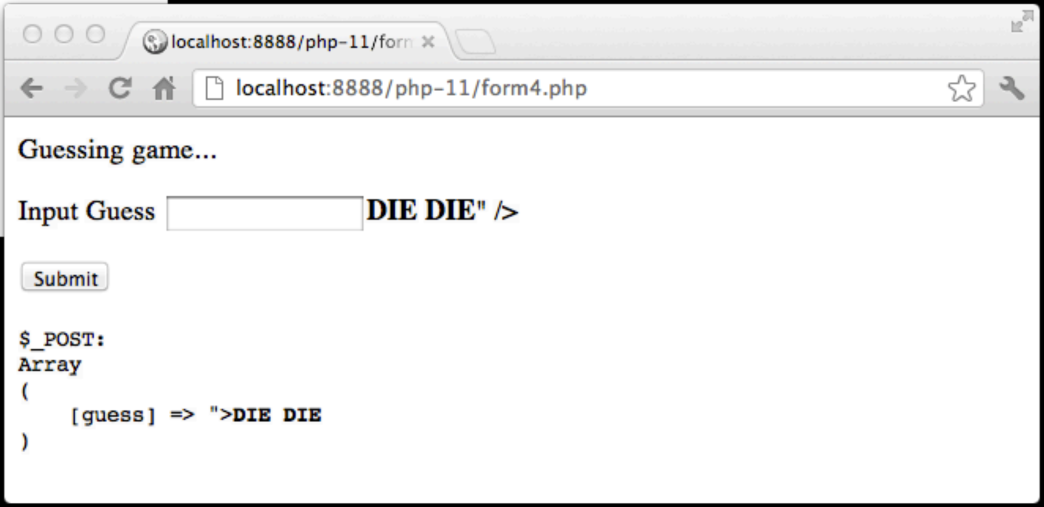
localhost:8888/php-11/form4.php

Guessing game...

Input Guess `"><b>DIE DIE</b>`

Submit

```
$_POST:
Array
(
)
```



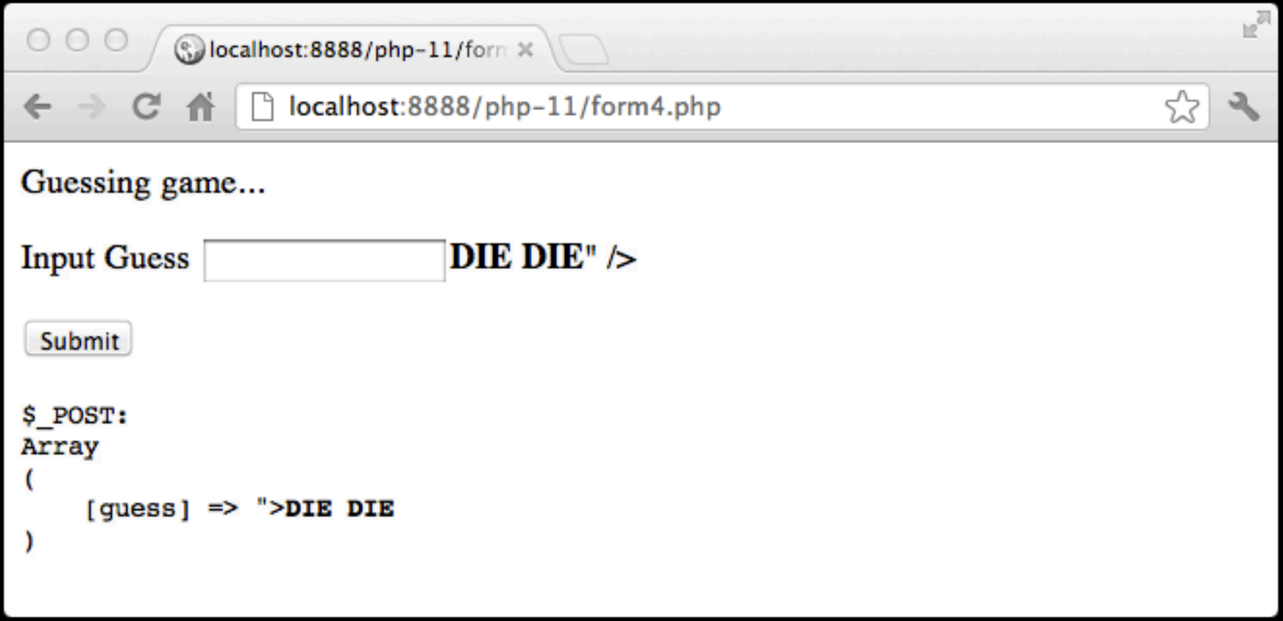
localhost:8888/php-11/form4.php

Guessing game...

Input Guess **DIE DIE" />**

Submit

```
$_POST:
Array
(
    [guess] => ">DIE DIE
)
```



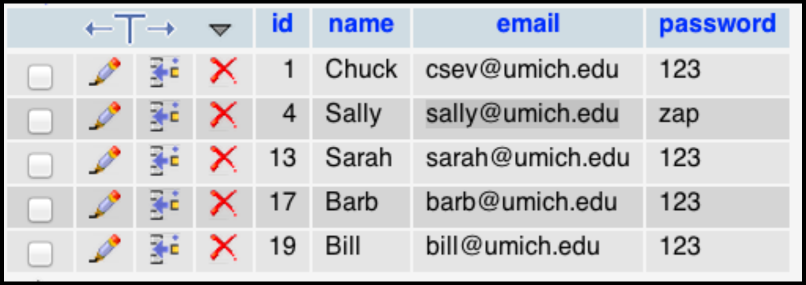
Guessing game...

Input Guess DIE DIE" />

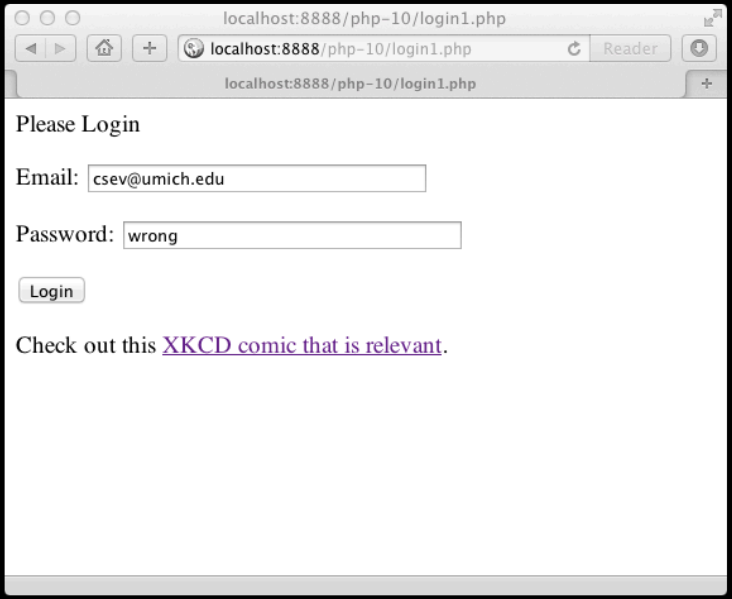
```
$_POST:
Array
(
    [guess] => ">DIE DIE
)
```

```
<form method="post">
   <p><label for="guess">Input Guess</label>
   <input type="text" name="guess" id="guess"
value=""><b>DIE DIE</b>"   /></p>
   <input type="submit"/>
</form>
```

What Could Go Wrong?



	id	name	email	password
	1	Chuck	csev@umich.edu	123
	4	Sally	sally@umich.edu	zap
	13	Sarah	sarah@umich.edu	123
	17	Barb	barb@umich.edu	123
	19	Bill	bill@umich.edu	123



localhost:8888/php-10/login1.php

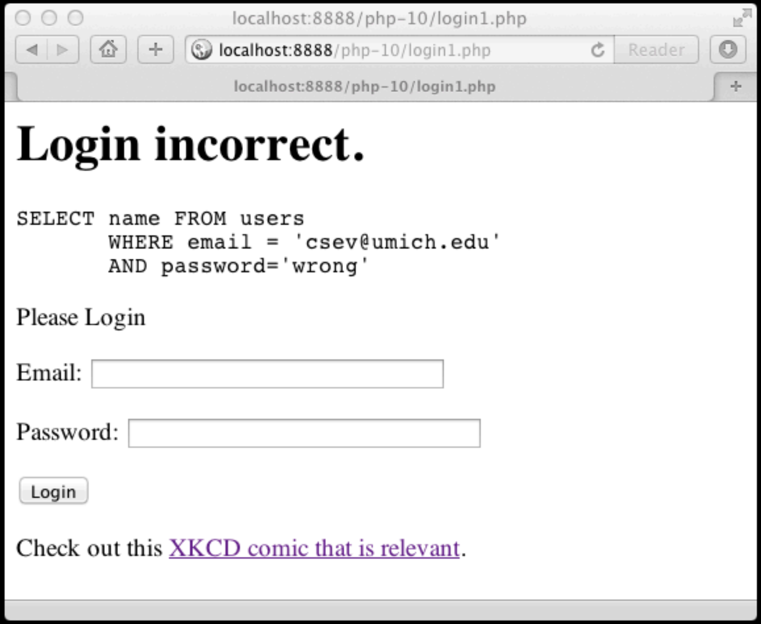
Please Login

Email: csev@umich.edu

Password: wrong

Login

Check out this XKCD comic that is relevant.



localhost:8888/php-10/login1.php

Login incorrect.

```
SELECT name FROM users
       WHERE email = 'csev@umich.edu'
       AND password='wrong'
```

Please Login

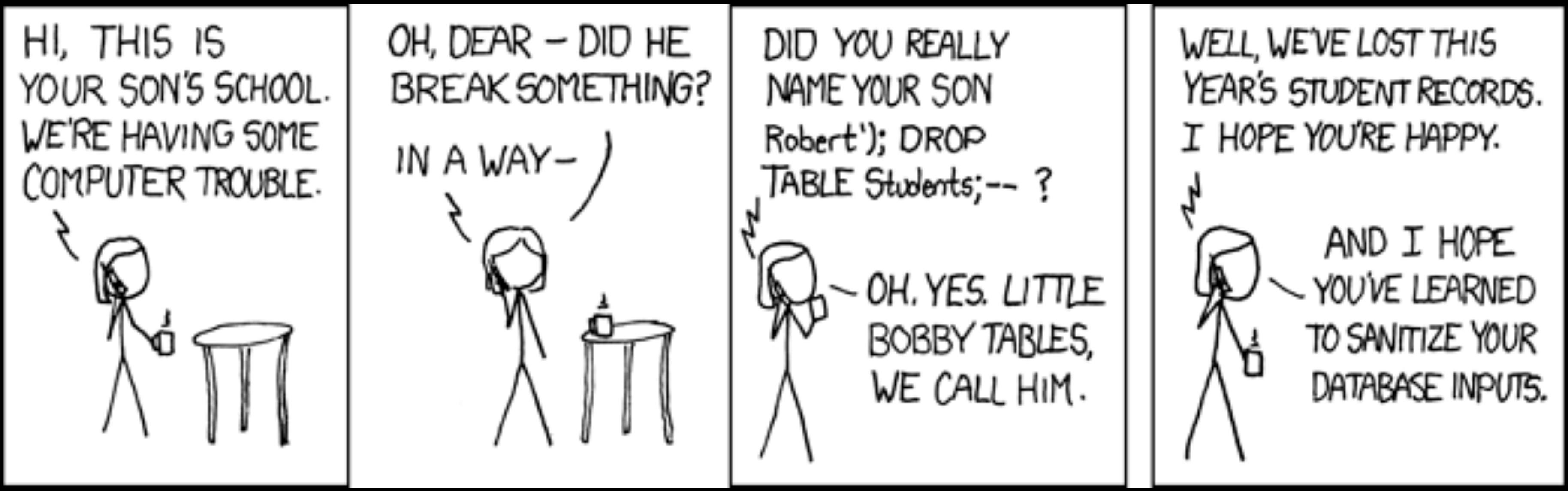
Email:

Password:

Login

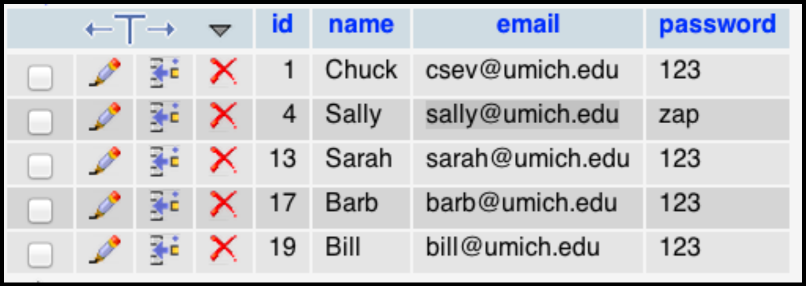
Check out this XKCD comic that is relevant.

login1.php

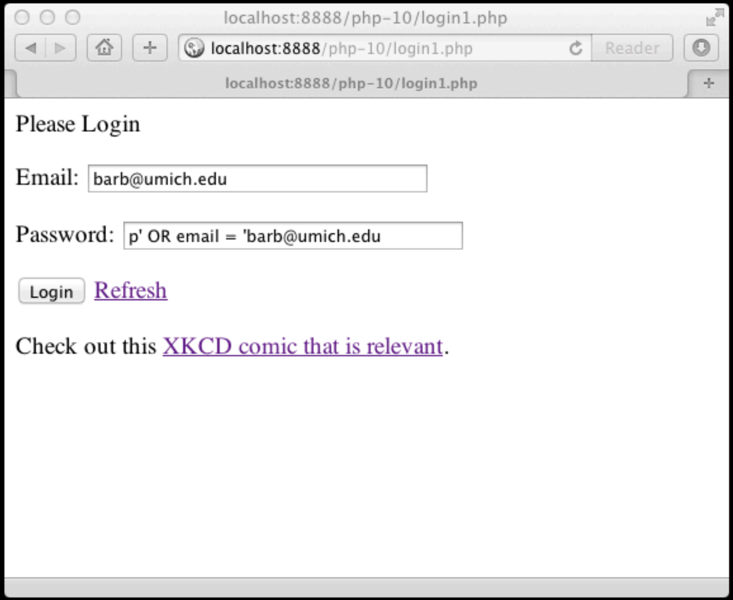

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

http://xkcd.com/327/

```
if ( isset($_POST['email']) && isset($_POST['password'])  ) {
   $e = $_POST['email'];
   $p = $_POST['password'];
   $sql = "SELECT name FROM users
       WHERE email = '$e'
       AND password = '$p'";

   $result = mysql_query($sql);
```



id	name	email	password
1	Chuck	csev@umich.edu	123
4	Sally	sally@umich.edu	zap
13	Sarah	sarah@umich.edu	123
17	Barb	barb@umich.edu	123
19	Bill	bill@umich.edu	123



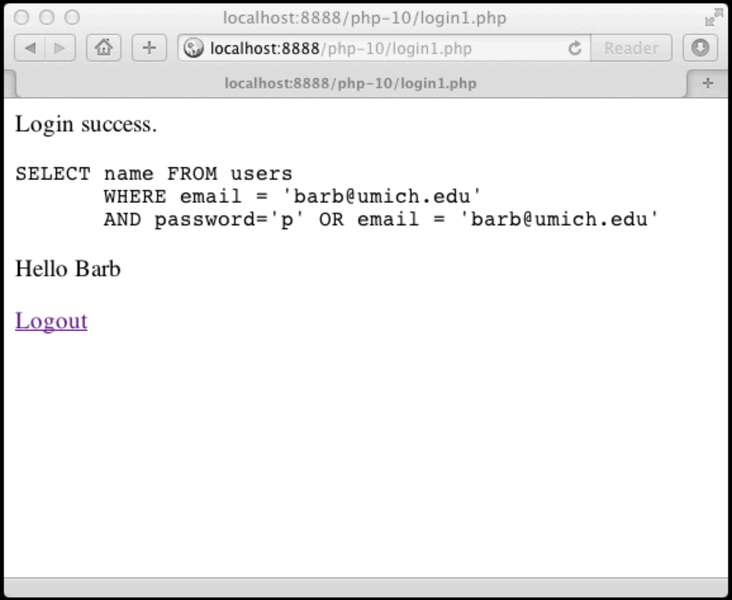
Please Login

Email: barb@umich.edu

Password: p' OR email = 'barb@umich.edu

Login Refresh

Check out this XKCD comic that is relevant.



Login success.

```
SELECT name FROM users
       WHERE email = 'barb@umich.edu'
       AND password='p' OR email = 'barb@umich.edu'
```

Hello Barb

Logout

login1.php

Security Lesson: SQL Injection

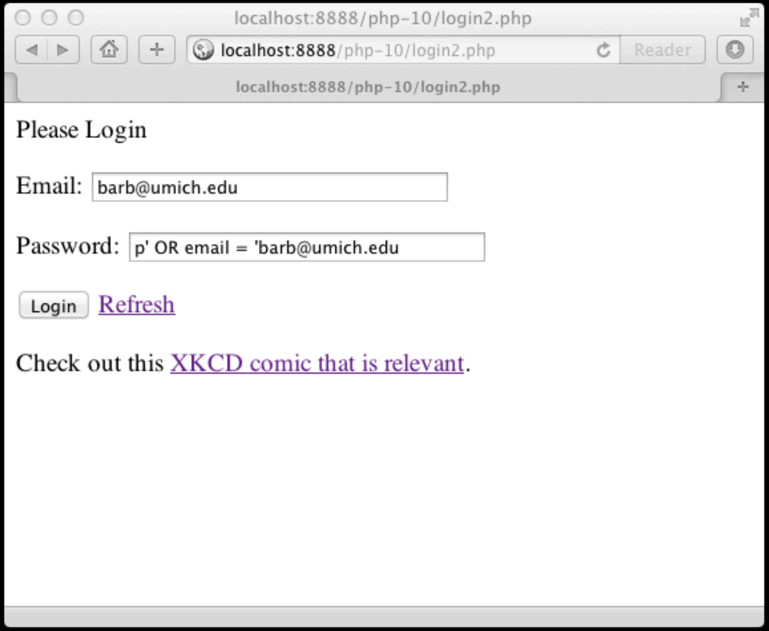
- NEVER EVER EVER EVER take values from the outside world and put them in an SQL string without using
- mysql_real_escape_string()

```
if ( isset($_POST['email']) && isset($_POST['password'])  ) {
   $e = mysql_real_escape_string($_POST['email']);
   $p = mysql_real_escape_string($_POST['password']);
   $sql = "SELECT name FROM users
       WHERE email = '$e'
       AND password = '$p'";

   $result = mysql_query($sql);
```

login2.php

login2.php

```
if ( isset($_POST['email']) && isset($_POST['password'])  ) {
   $e = mysql_real_escape_string($_POST['email']);
   $p = mysql_real_escape_string($_POST['password']);
   $sql = "SELECT name FROM users
       WHERE email = '$e'
       AND password = '$p'";

   $result = mysql_query($sql);
```

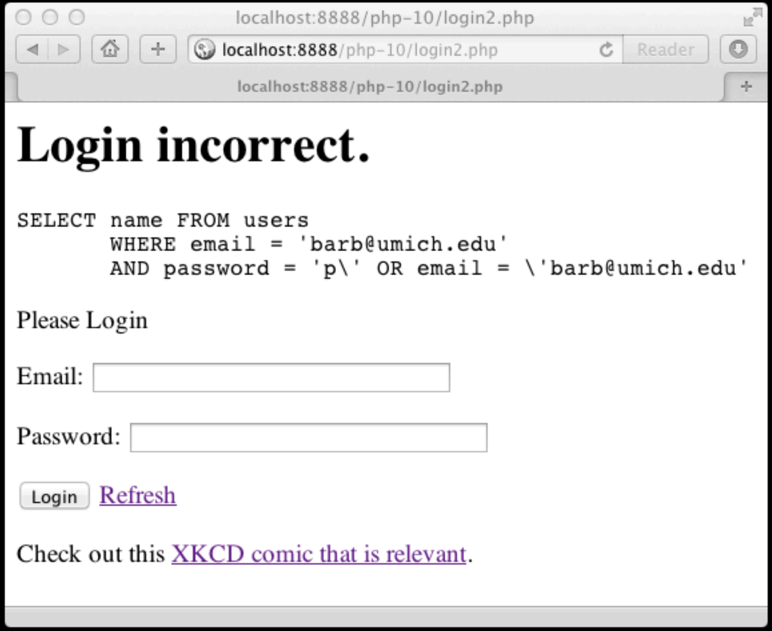


Please Login	
Email:	barb@umich.edu
Password:	p' OR email = 'barb@umich.edu

Login Refresh

Check out this XKCD comic that is relevant.

login2.php



Login incorrect.

```
SELECT name FROM users
       WHERE email = 'barb@umich.edu'
       AND password = 'p\' OR email = \'barb@umich.edu'
```

Please Login

Email:

Password:

Login Refresh

Check out this XKCD comic that is relevant.

Rescue: mysql_real_escape_string()

```
if ( isset($_POST['email']) &&
isset($_POST['password'])  ) {
   $e = mysql_real_escape_string($_POST['email']);
   $p = mysql_real_escape_string($_POST['password']);
   $sql = "SELECT name FROM users
       WHERE email = '$e'
       AND password = '$p'";

   $result = mysql_query($sql);
```

Escapes special characters in the string, taking into account the current character set of the connection so that it is safe to place it in a mysql_query(). If binary data is to be inserted, this function must be used.

http://php.net/manual/en/function.mysql-real-escape-string.php

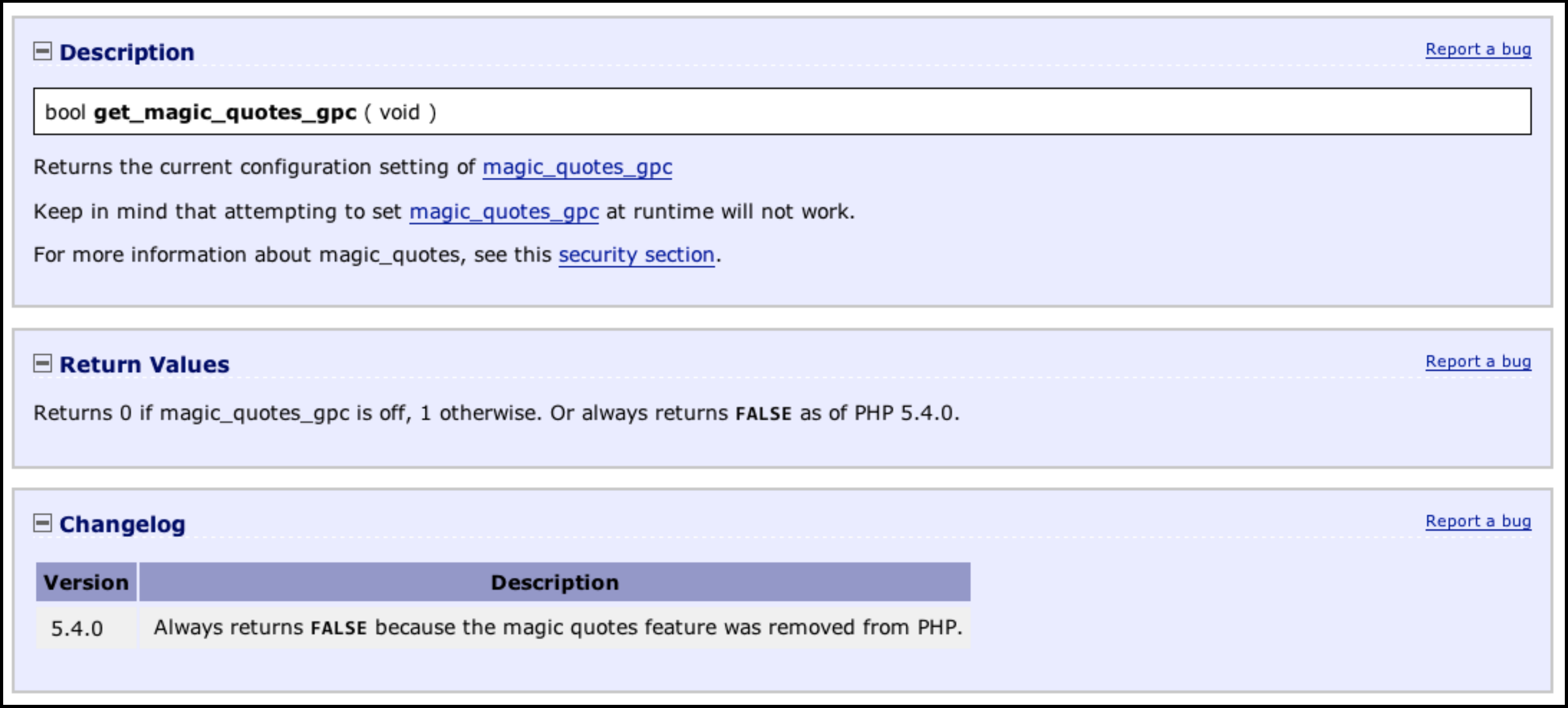
login3.php

Rescue #2: Use PDO

```
if ( isset($_POST['email']) && isset($_POST['password'])  ) {
    echo("Handling POST data...\n");
    $sql = "SELECT name FROM users
        WHERE email = :em AND password = :pw";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':em' => $_POST['email'],
        ':pw' => $_POST['password']));
    $row = $stmt->fetch(PDO::FETCH_ASSOC);
```

PDO uses a pattern called 'prepared statements' that only have named placeholders in the SQL. When the statement is executed, the placeholders get replaced with the actual strings and everything is automatically escaped!

A Kind of Icky Digression...



Description Report a bug

```
bool get_magic_quotes_gpc ( void )
```

Returns the current configuration setting of magic_quotes_gpc

Keep in mind that attempting to set magic_quotes_gpc at runtime will not work.

For more information about magic_quotes, see this security section.

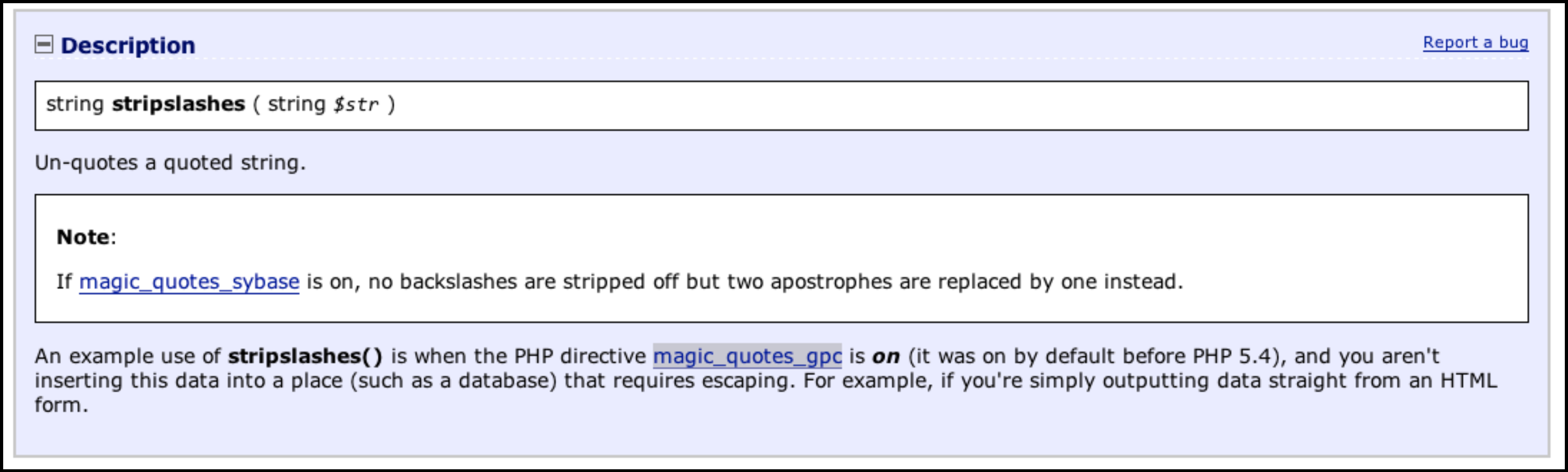
Return Values Report a bug

Returns 0 if magic_quotes_gpc is off, 1 otherwise. Or always returns **FALSE** as of PHP 5.4.0.

Changelog Report a bug

Version	Description
5.4.0	Always returns **FALSE** because the magic quotes feature was removed from PHP.

http://php.net/manual/en/function.get-magic-quotes-gpc.php



Description Report a bug

string **stripslashes** (string *$str*)

Un-quotes a quoted string.

Note:

If magic_quotes_sybase is on, no backslashes are stripped off but two apostrophes are replaced by one instead.

An example use of **stripslashes()** is when the PHP directive magic_quotes_gpc is ***on*** (it was on by default before PHP 5.4), and you aren't inserting this data into a place (such as a database) that requires escaping. For example, if you're simply outputting data straight from an HTML form.

http://php.net/manual/en/function.stripslashes.php

```
echo("Handling POST data...\n");
$e = $_POST['email'];
$p = $_POST['password'];
echo("Password: $p\n");
if ( get_magic_quotes_gpc() ) {
     echo("Magic quotes are on...\n");
     $e = stripslashes($e);
     $p = stripslashes($p);
     echo("Raw Password: $p\n");
} else {
     echo("Magic quotes are off...\n");
}

$sql = "SELECT name FROM users
    WHERE email = '$e'
    AND password = '$p'";
```

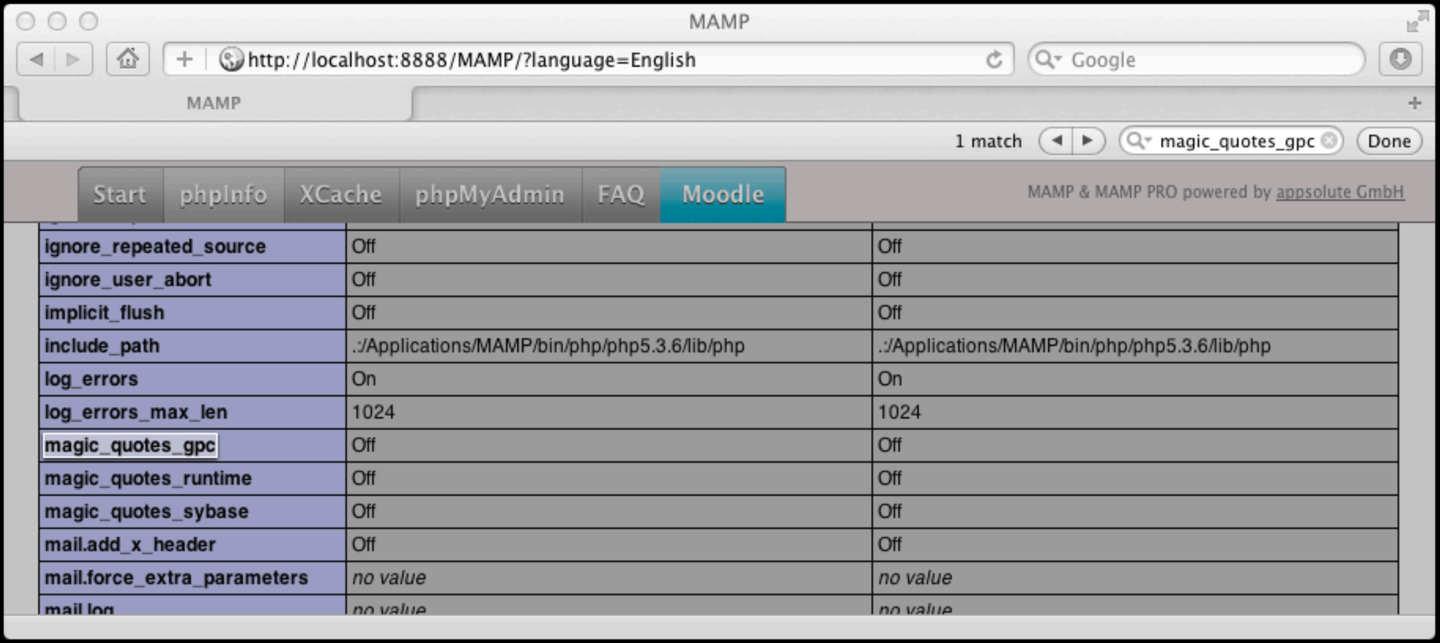
Not needed after php 5.4...

login1.php

```
echo("Handling POST data...\n");
$e = mysql_real_escape_string($_POST['email']);
$p = mysql_real_escape_string($_POST['password']);

$sql = "SELECT name FROM users
    WHERE email = '$e'
    AND password = '$p'";
```

mysql_real_escape_string() is aware of get_magic_quotes_gpc()

login2.php


MAMP
http://localhost:8888/MAMP/?language=English
Google
1 match
magic_quotes_gpc
Done
Start
phpInfo
XCache
phpMyAdmin
FAQ
Moodle
MAMP & MAMP PRO powered by appsolute GmbH

ignore_repeated_source	Off	Off
ignore_user_abort	Off	Off
implicit_flush	Off	Off
include_path	.:/Applications/MAMP/bin/php/php5.3.6/lib/php	.:/Applications/MAMP/bin/php/php5.3.6/lib/php
log_errors	On	On
log_errors_max_len	1024	1024
magic_quotes_gpc	Off	Off
magic_quotes_runtime	Off	Off
magic_quotes_sybase	Off	Off
mail.add_x_header	Off	Off
mail.force_extra_parameters	no value	no value
mail.log	no value	no value

MAMP: bin/php/php5.3.2/conf/php.ini

```
magic_quotes_gpc = Off
```

Not needed after php 5.4...

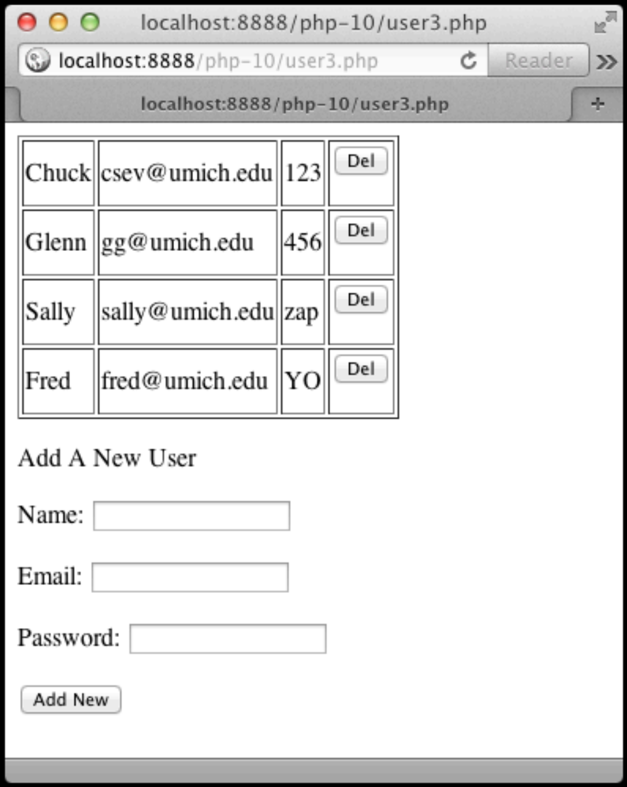
CRUD!

CRUD Pattern

- When we store things in database tables we generally need
  - Create - Insert a new row
  - Read - Read existing row(s)
  - Update - Change some values of a record
  - Delete - Delete a record
- So far we have done 3/4 of CRUD

Our Program is a little Ugly

- Usually we create several screens
  - Add new row
  - View all rows (paging)
  - View single row
  - Edit single row
  - Delete a row



Chuck	csev@umich.edu	123	Del
Glenn	gg@umich.edu	456	Del
Sally	sally@umich.edu	zap	Del
Fred	fred@umich.edu	YO	Del

Add A New User

Name:

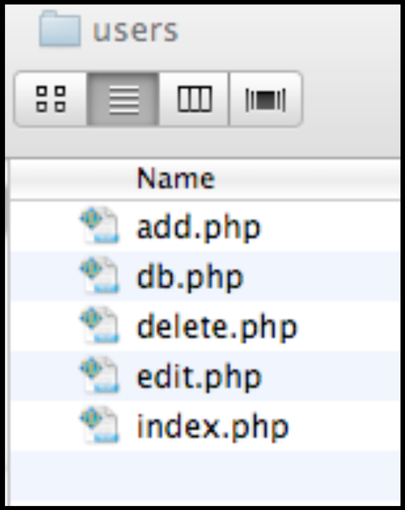
Email:

Password:

Add New

Five Separate Files

- index.php - Main list and links to other files
- add.php - Add a new entry
- delete.php - Delete an entry
- edit.php - Edit existing
- view.php (if index.php needed a detail view)



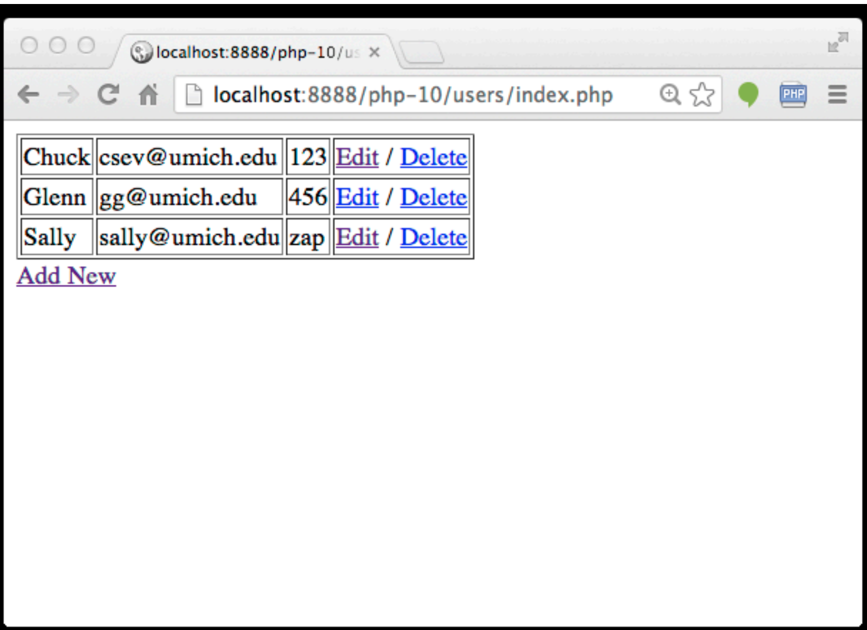
add.php
db.php
delete.php
edit.php
index.php

index.php

```
<?php
require_once "db.php";
session_start();
?>
<html>
<head></head><body>
<?php
if ( isset($_SESSION['error']) ) {
    echo '<p style="color:red">'.$_SESSION['error']."</p>\n";
    unset($_SESSION['error']);
}
if ( isset($_SESSION['success']) ) {
    echo '<p style="color:green">'.$_SESSION['success']."</p>\n";
    unset($_SESSION['success']);
}
echo('<table border="1">'."\n");
```

index.php

```
echo('<table border="1">'."\n");
$stmt = $pdo->query("SELECT name, email, password, id FROM users");
while ( $row = $stmt->fetch(PDO::FETCH_ASSOC) ) {
    echo "<tr><td>";
    echo($row['name']);
    echo("</td><td>");
    echo($row['email']);
    echo("</td><td>");
    echo($row['password']);
    echo("</td><td>");
    echo('<a href="edit.php?id='.htmlentities($row['id']).'">Edit</a> / ');
    echo('<a href="delete.php?id='.htmlentities($row['id']).'">Delete</a>');
    echo("\n</form>\n");
    echo("</td></tr>\n");
}
?>
</table>
<a href="add.php">Add New</a>
```


localhost:8888/php-10/users/index.php

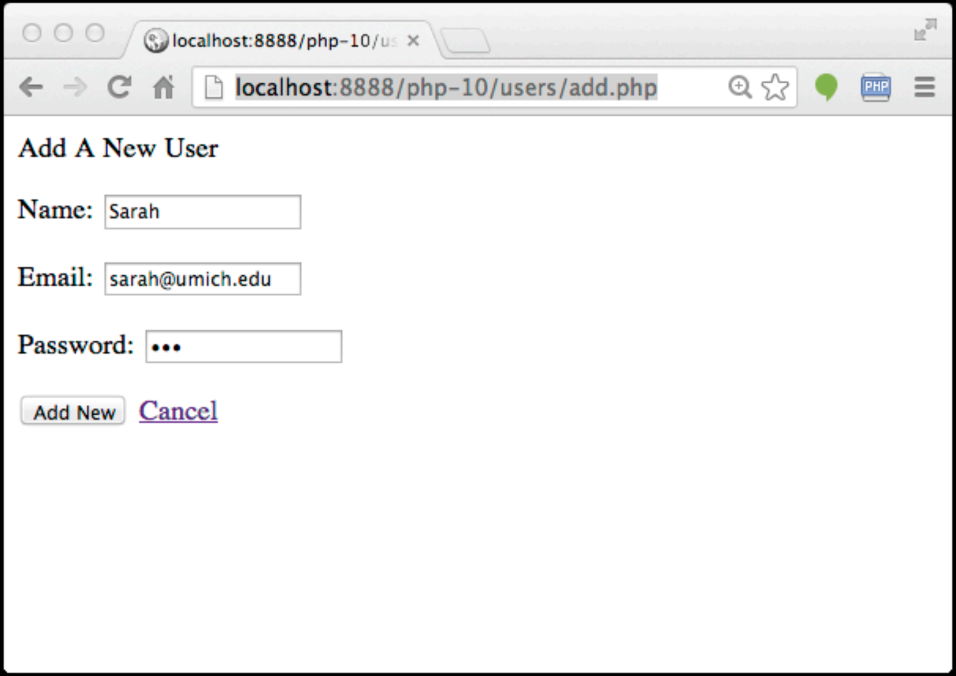
Chuck	csev@umich.edu	123	Edit / Delete
Glenn	gg@umich.edu	456	Edit / Delete
Sally	sally@umich.edu	zap	Edit / Delete

Add New

```
<tr><td>Chuck</td><td>csev@umich.edu</td><td>123</td><td>
<a href="edit.php?id=1">Edit</a> /
<a href="delete.php?id=1">Delete</a></td></tr>

<tr><td>Glenn</td><td>gg@umich.edu</td><td>456</td><td>
<a href="edit.php?id=2">Edit</a> /
<a href="delete.php?id=2">Delete</a></td></tr>
```

add.php

```
<?php
require_once "db.php";
session_start();

if ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password'])) {
    $sql = "INSERT INTO users (name, email, password)
              VALUES (:name, :email, :password)";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password']));
   $_SESSION['success'] = 'Record Added';
   header( 'Location: index.php' ) ;
   return;
}
?>
<p>Add A New User</p>
<form method="post">
<p>Name:
<input type="text" name="name"></p>
<p>Email:
<input type="text" name="email"></p>
<p>Password:
<input type="password" name="password"></p>
<p><input type="submit" value="Add New"/>
<a href="index.php">Cancel</a></p>
</form>
```

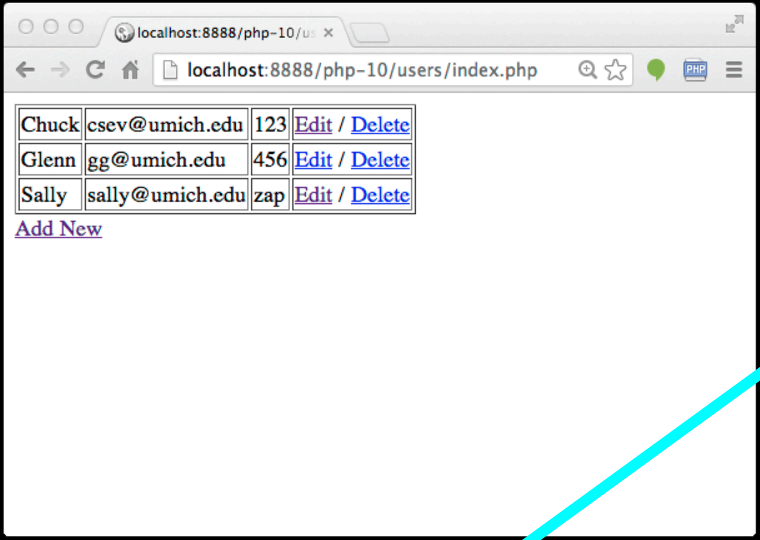


Add A New User

Name: Sarah

Email: sarah@umich.edu

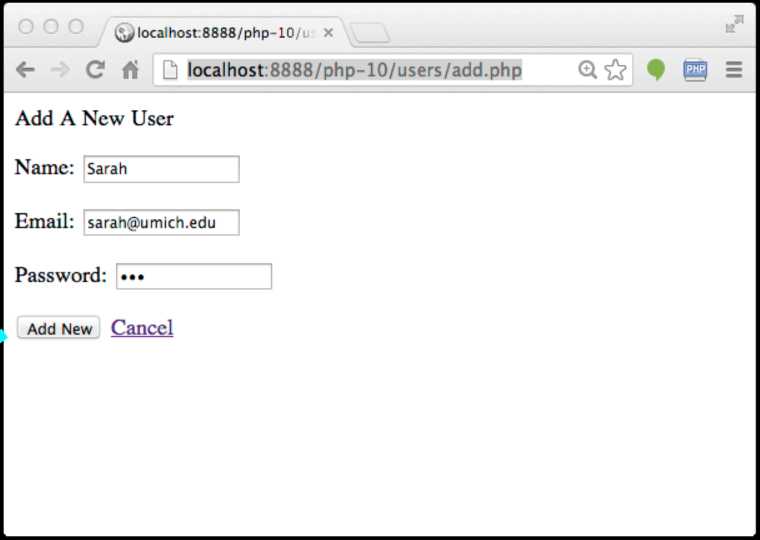
Password: •••



localhost:8888/php-10/users/index.php

Chuck	csev@umich.edu	123	Edit / Delete
Glenn	gg@umich.edu	456	Edit / Delete
Sally	sally@umich.edu	zap	Edit / Delete

Add New



localhost:8888/php-10/users/add.php

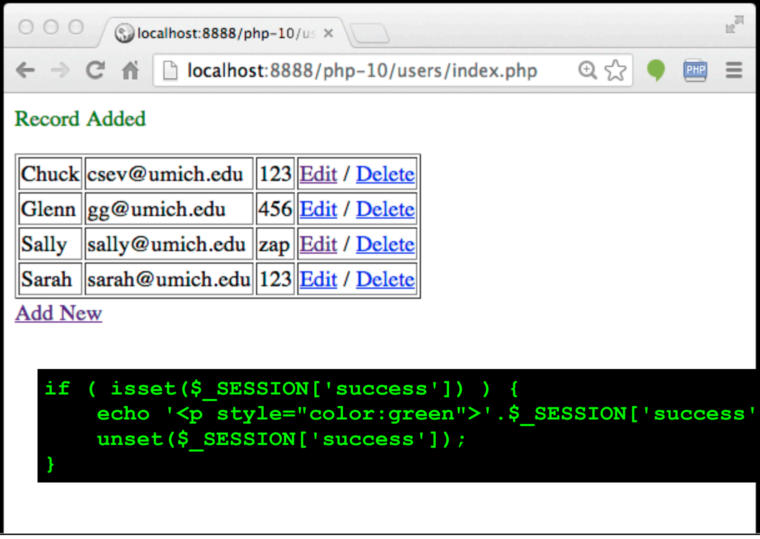
Add A New User

Name: Sarah

Email: sarah@umich.edu

Password: •••

```
if ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password'])) {
    $sql = "INSERT INTO users (name, email, password)
              VALUES (:name, :email, :password)";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password']));
   $_SESSION['success'] = 'Record Added';
   header( 'Location: index.php' ) ;
   return;
```



localhost:8888/php-10/users/index.php

Record Added

Chuck	csev@umich.edu	123	Edit / Delete
Glenn	gg@umich.edu	456	Edit / Delete
Sally	sally@umich.edu	zap	Edit / Delete
Sarah	sarah@umich.edu	123	Edit / Delete

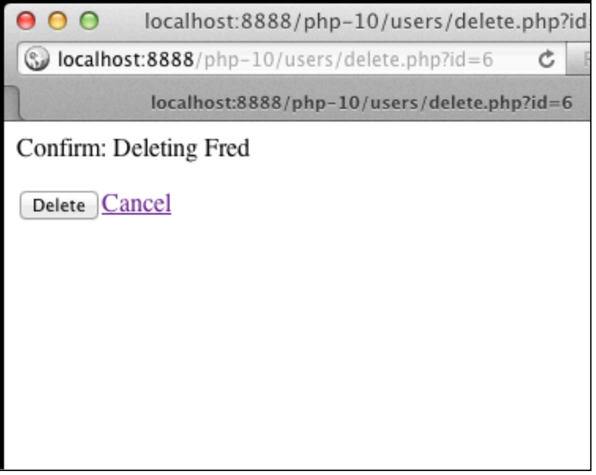
Add New

```
if ( isset($_SESSION['success']) ) {
    echo '<p style="color:green">'.$_SESSION['success']."</p>\n";
    unset($_SESSION['success']);
}
```

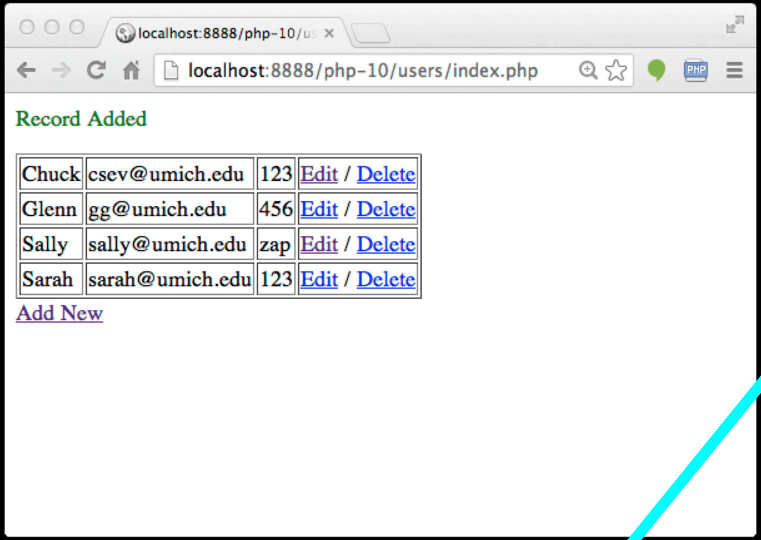
delete.php

```php
<?php
require_once "db.php";
session_start();

if ( isset($_POST['delete']) && isset($_POST['id']) ) {
    $sql = "DELETE FROM users WHERE id = :zip";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['id']));
    $_SESSION['success'] = 'Record deleted';
    header( 'Location: index.php' ) ;
    return;
}

$stmt = $pdo->prepare("SELECT name, id FROM users where id = :xyz");
$stmt->execute(array(":xyz" => $_GET['id']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
if ( $row === false ) {
    $_SESSION['error'] = 'Bad value for id';
    header( 'Location: index.php' ) ;
    return;
}

echo "<p>Confirm: Deleting ".$row['name']."</p>\n";

echo('<form method="post"><input type="hidden" ');
echo('name="id" value="'.$row['id'].'">'."\n");
echo('<input type="submit" value="Delete" name="delete">');
echo('<a href="index.php">Cancel</a>');
echo("\n</form>\n");
?>
```

Don't alter data in a GET.


localhost:8888/php-10/users/delete.php?id
localhost:8888/php-10/users/delete.php?id=6
localhost:8888/php-10/users/delete.php?id=6
Delete
Cancel

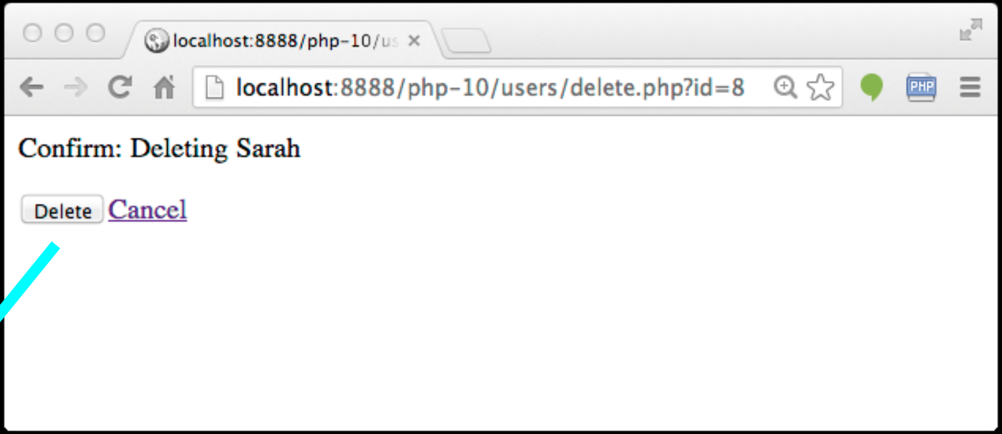
Confirm: Deleting Fred



Record Added

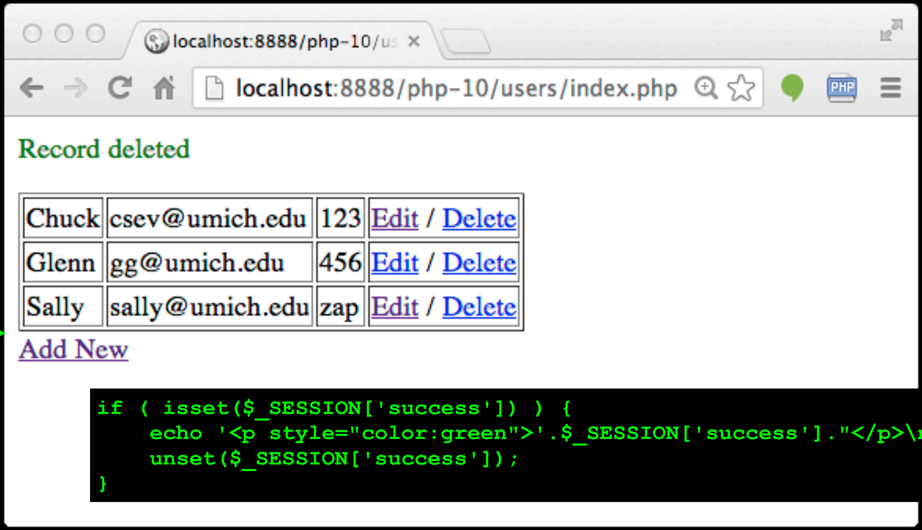
Chuck	csev@umich.edu	123	Edit / Delete
Glenn	gg@umich.edu	456	Edit / Delete
Sally	sally@umich.edu	zap	Edit / Delete
Sarah	sarah@umich.edu	123	Edit / Delete

Add New



Confirm: Deleting Sarah

```
if ( isset($_POST['delete']) && isset($_POST['id']) ) {
    $sql = "DELETE FROM users WHERE id = :zip";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['id']));
    $_SESSION['success'] = 'Record deleted';
    header( 'Location: index.php' ) ;
    return;
}
```



Record deleted

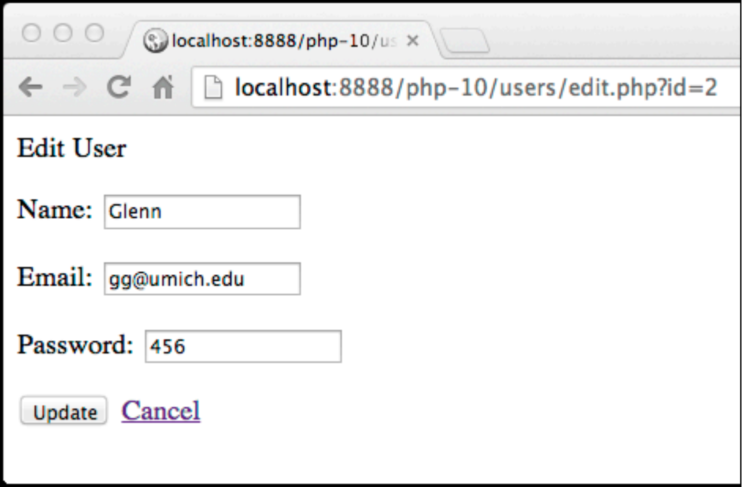
Chuck	csev@umich.edu	123	Edit / Delete
Glenn	gg@umich.edu	456	Edit / Delete
Sally	sally@umich.edu	zap	Edit / Delete

Add New

```
if ( isset($_SESSION['success']) ) {
    echo '<p style="color:green">'.$_SESSION['success']."</p>\n";
    unset($_SESSION['success']);
}
```

edit.php

```
<?php
require_once "db.php";
session_start();

if ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password']) && isset($_POST['id']) ) {
    $sql = "UPDATE users SET name = :name,
            email = :email, password = :password
            WHERE id = :id";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password'],
        ':id' => $_POST['id']));
    $_SESSION['success'] = 'Record updated';
    header( 'Location: index.php' ) ;
    return;
}

$stmt = $pdo->prepare("SELECT * FROM users where id = :xyz");
$stmt->execute(array(":xyz" => $_GET['id']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
if ( $row === false ) {
    $_SESSION['error'] = 'Bad value for id';
    header( 'Location: index.php' ) ;
    return;
}
$n = htmlentities($row['name']);
$e = htmlentities($row['email']);
$p = htmlentities($row['password']);
$id = htmlentities($row['id']);
```


localhost:8888/php-10/users/edit.php?id=2

Edit User

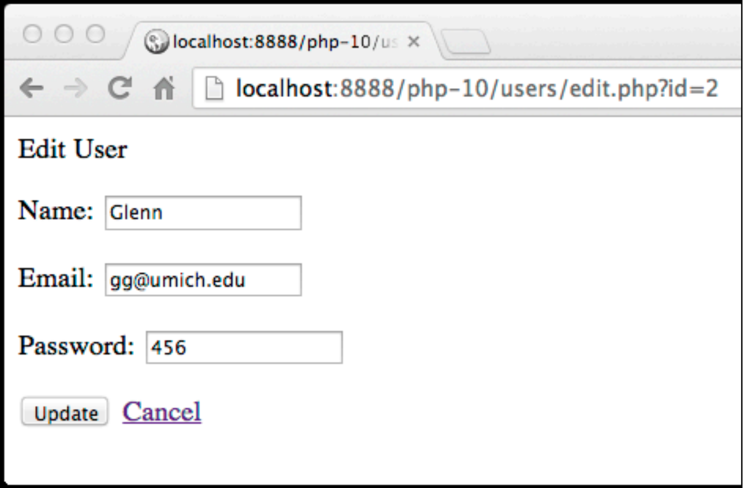
Name: Glenn

Email: gg@umich.edu

Password: 456

edit.php

```
$n = htmlentities($row['name']);
$e = htmlentities($row['email']);
$p = htmlentities($row['password']);
$id = htmlentities($row['id']);

echo <<< _END
<html>
<head></head><body>
<p>Edit User</p>
<form method="post">
<p>Name:
<input type="text" name="name" value="$n"></p>
<p>Email:
<input type="text" name="email" value="$e"></p>
<p>Password:
<input type="text" name="password" value="$p"></p>
<input type="hidden" name="id" value="$id">
<p><input type="submit" value="Update"/>
<a href="index.php">Cancel</a></p>
</form>
</body>
_END
```

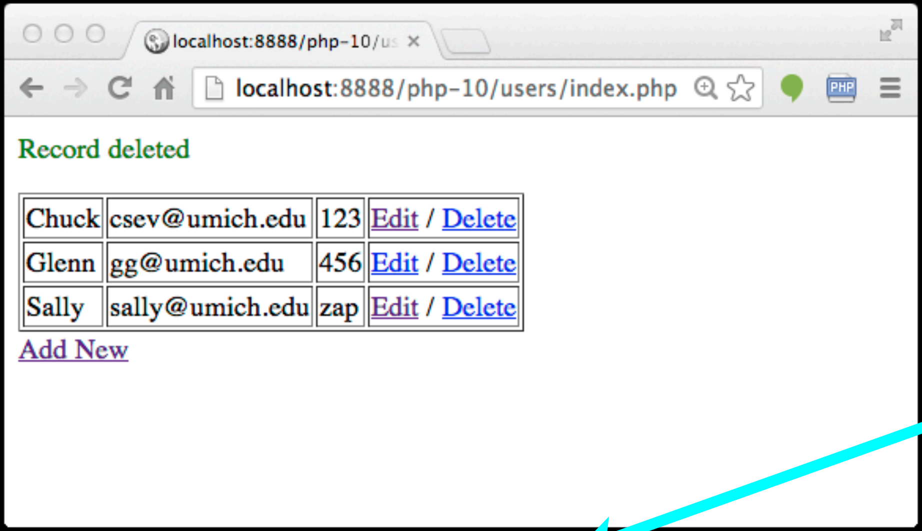


Edit User

Name: Glenn

Email: gg@umich.edu

Password: 456

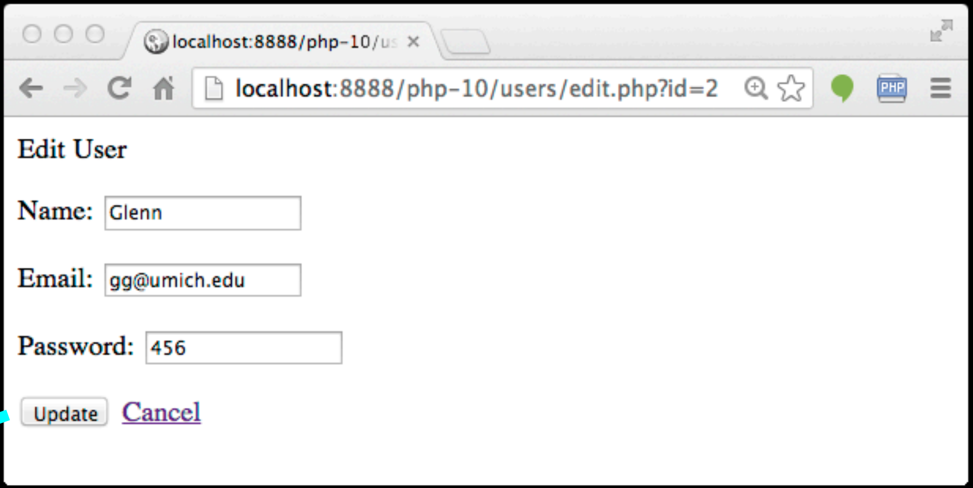


localhost:8888/php-10/users/index.php

Record deleted

Chuck	csev@umich.edu	123	Edit / Delete
Glenn	gg@umich.edu	456	Edit / Delete
Sally	sally@umich.edu	zap	Edit / Delete

Add New



localhost:8888/php-10/users/edit.php?id=2

Edit User

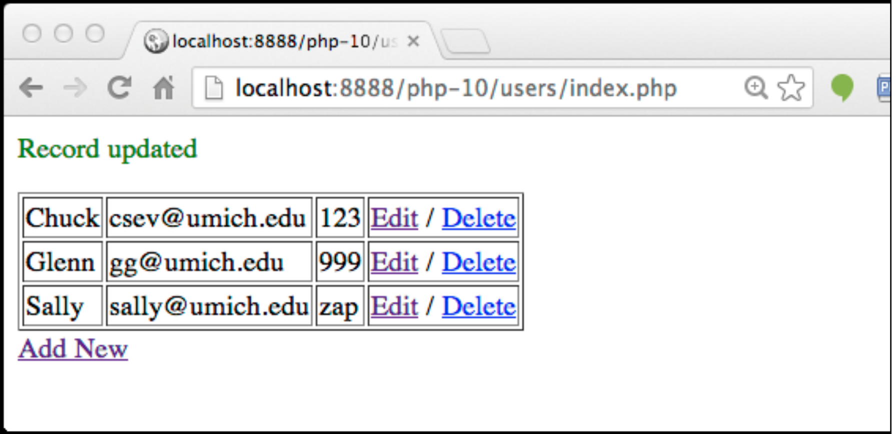
Name: Glenn

Email: gg@umich.edu

Password: 456

Update Cancel

```
if ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password']) && isset($_POST['id']) ) {
    $sql = "UPDATE users SET name = :name,
            email = :email, password = :password
            WHERE id = :id";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password'],
        ':id' => $_POST['id']));
    $_SESSION['success'] = 'Record updated';
    header( 'Location: index.php' ) ;
    return;
}
```



localhost:8888/php-10/users/index.php

Record updated

Chuck	csev@umich.edu	123	Edit / Delete
Glenn	gg@umich.edu	999	Edit / Delete
Sally	sally@umich.edu	zap	Edit / Delete

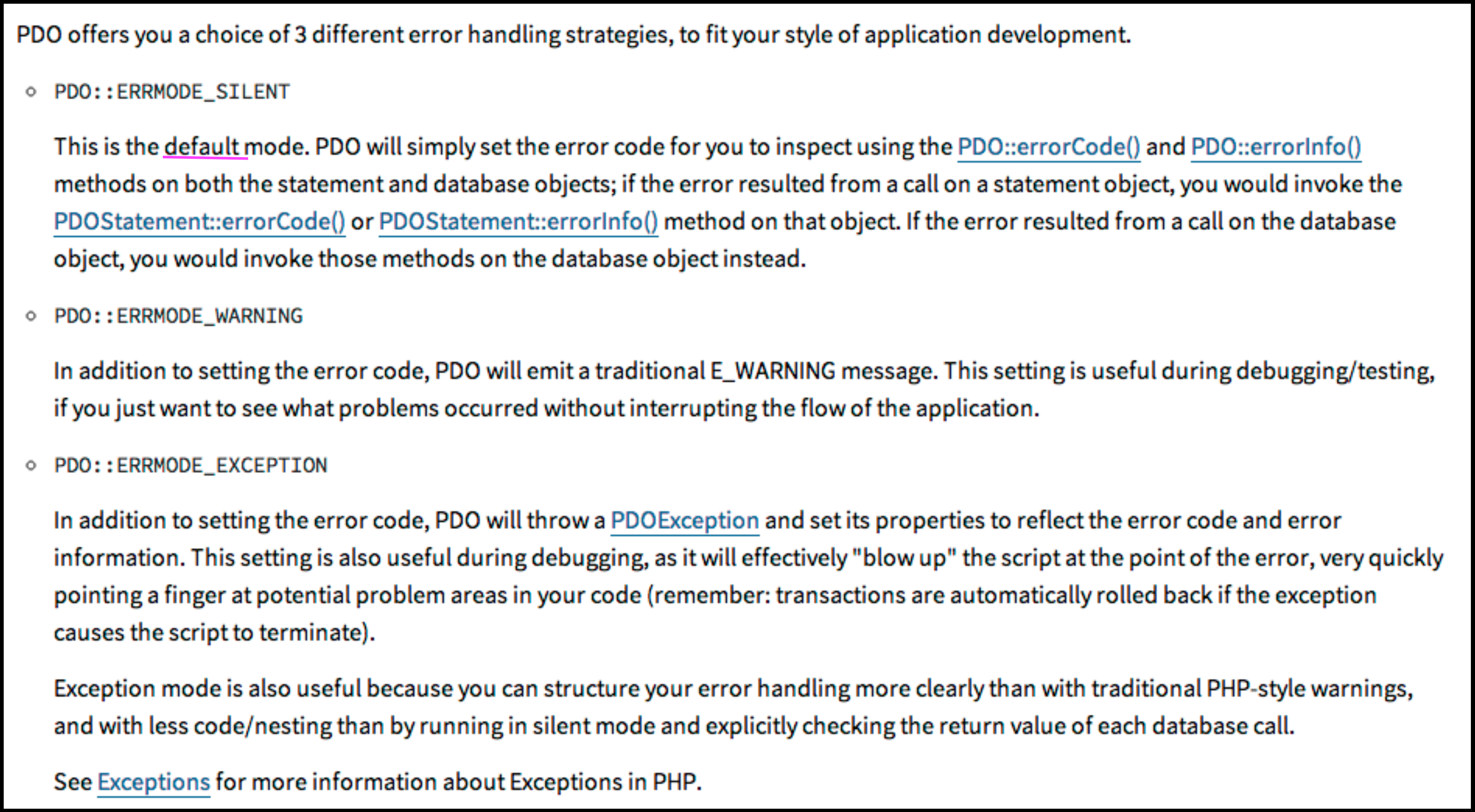
Add New

edit.php

What Could Go Wrong?

errors/edit1.php

```
$stmt = $pdo->prepare("SELECT * FROM users where id = :xyz");
$stmt->execute(array(":pizza" => $_GET['id']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
if ( $row === false ) {
    $_SESSION['error'] = 'Bad value for id';
    header( 'Location: index.php' ) ;
    return;
}
```



PDO offers you a choice of 3 different error handling strategies, to fit your style of application development.

- `PDO::ERRMODE_SILENT`

  This is the default mode. PDO will simply set the error code for you to inspect using the PDO::errorCode() and PDO::errorInfo() methods on both the statement and database objects; if the error resulted from a call on a statement object, you would invoke the PDOStatement::errorCode() or PDOStatement::errorInfo() method on that object. If the error resulted from a call on the database object, you would invoke those methods on the database object instead.

- `PDO::ERRMODE_WARNING`

  In addition to setting the error code, PDO will emit a traditional E_WARNING message. This setting is useful during debugging/testing, if you just want to see what problems occurred without interrupting the flow of the application.

- `PDO::ERRMODE_EXCEPTION`

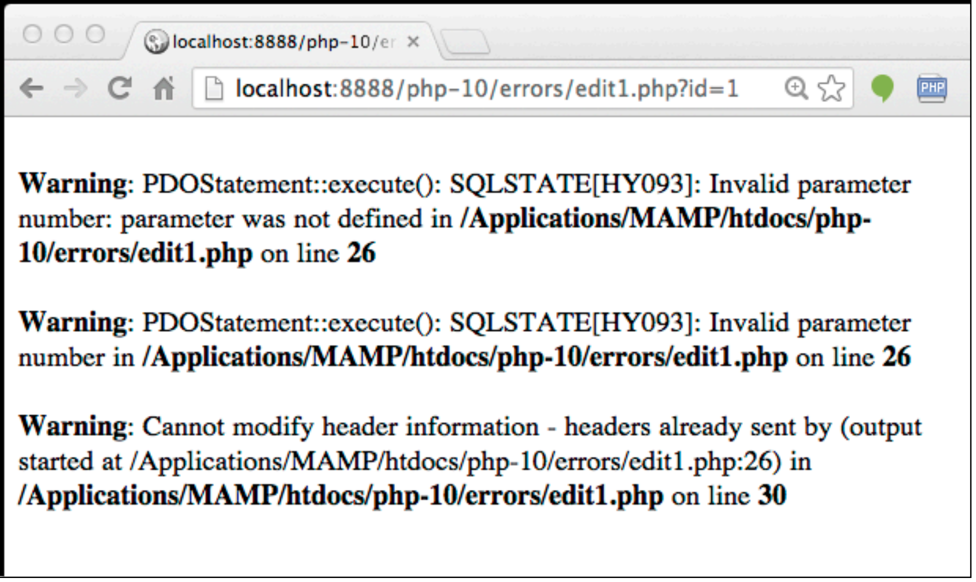
  In addition to setting the error code, PDO will throw a PDOException and set its properties to reflect the error code and error information. This setting is also useful during debugging, as it will effectively "blow up" the script at the point of the error, very quickly pointing a finger at potential problem areas in your code (remember: transactions are automatically rolled back if the exception causes the script to terminate).

  Exception mode is also useful because you can structure your error handling more clearly than with traditional PHP-style warnings, and with less code/nesting than by running in silent mode and explicitly checking the return value of each database call.

  See Exceptions for more information about Exceptions in PHP.

http://php.net/manual/en/pdo.error-handling.php

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);

$stmt = $pdo->prepare("SELECT * FROM users where id = :xyz");
$stmt->execute(array(":pizza" => $_GET['id']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
if ( $row === false ) {
    $_SESSION['error'] = 'Bad value for id';
    header( 'Location: index.php' ) ;
    return;
}
```



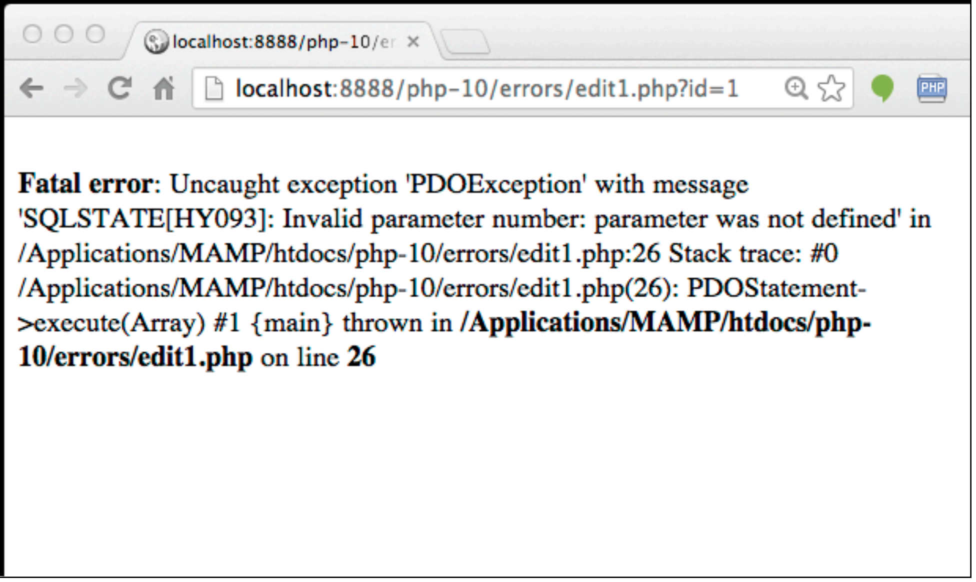
Warning: PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number: parameter was not defined in **/Applications/MAMP/htdocs/php-10/errors/edit1.php** on line **26**

Warning: PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number in **/Applications/MAMP/htdocs/php-10/errors/edit1.php** on line **26**

Warning: Cannot modify header information - headers already sent by (output started at /Applications/MAMP/htdocs/php-10/errors/edit1.php:26) in **/Applications/MAMP/htdocs/php-10/errors/edit1.php** on line **30**

errors/edit1.php

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$stmt = $pdo->prepare("SELECT * FROM users where id = :xyz");
$stmt->execute(array(":pizza" => $_GET['id']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
if ( $row === false ) {
    $_SESSION['error'] = 'Bad value for id';
    header( 'Location: index.php' ) ;
    return;
}
```

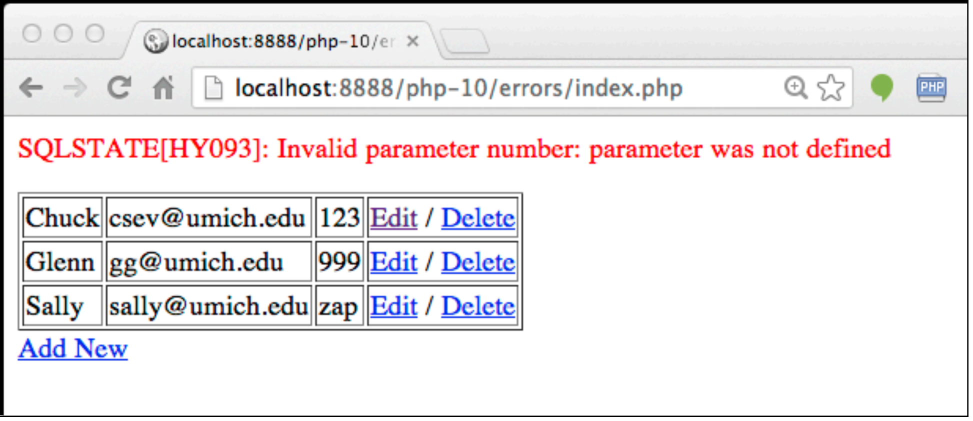


Fatal error: Uncaught exception 'PDOException' with message 'SQLSTATE[HY093]: Invalid parameter number: parameter was not defined' in /Applications/MAMP/htdocs/php-10/errors/edit1.php:26 Stack trace: #0 /Applications/MAMP/htdocs/php-10/errors/edit1.php(26): PDOStatement->execute(Array) #1 {main} thrown in **/Applications/MAMP/htdocs/php-10/errors/edit1.php** on line **26**

errors/edit2.php

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

try {
    $stmt = $pdo->prepare("SELECT * FROM users where id = :xyz");
    $stmt->execute(array(":pizza" => $_GET['id']));
} catch (Exception $ex ) {
    $_SESSION['error'] = $ex->getMessage();
    header( 'Location: index.php' ) ;
    return;
}
$row = $stmt->fetch(PDO::FETCH_ASSOC);

if ( $row === false ) {
    $_SESSION['error'] = 'Bad value for id';
    header( 'Location: index.php' ) ;
    return;
}
```

errors/edit3.php


localhost:8888/php-10/errors/index.php

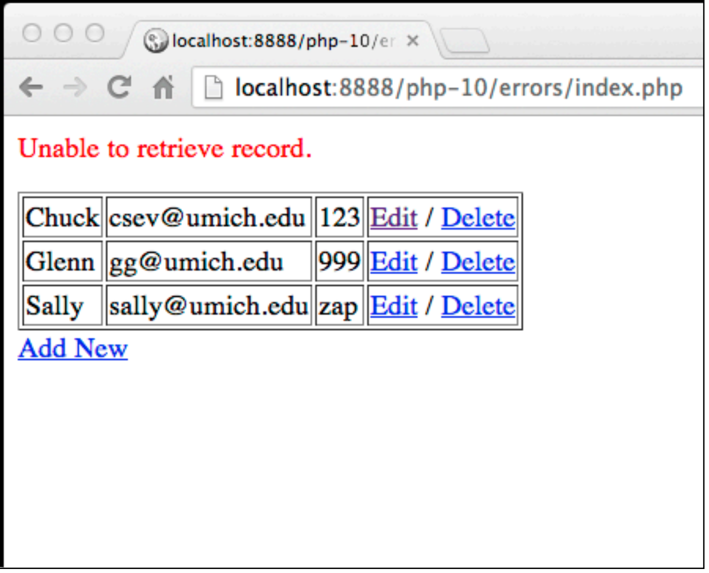
SQLSTATE[HY093]: Invalid parameter number: parameter was not defined

Chuck	csev@umich.edu	123	Edit / Delete
Glenn	gg@umich.edu	999	Edit / Delete
Sally	sally@umich.edu	zap	Edit / Delete

Add New

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

try {
    $stmt = $pdo->prepare("SELECT * FROM users where id = :xyz");
    $stmt->execute(array(":pizza" => $_GET['id']));
} catch (Exception $ex ) {
    error_log("edit.php, SQL error=".$ex->getMessage());
    $_SESSION['error'] = "Unable to retrieve record.";
    header( 'Location: index.php' ) ;
    return;
}
$row = $stmt->fetch(PDO::FETCH_ASSOC);

if ( $row === false ) {
    $_SESSION['error'] = 'Bad value for id';
    header( 'Location: index.php' ) ;
    return;
}
```

errors/edit4.php



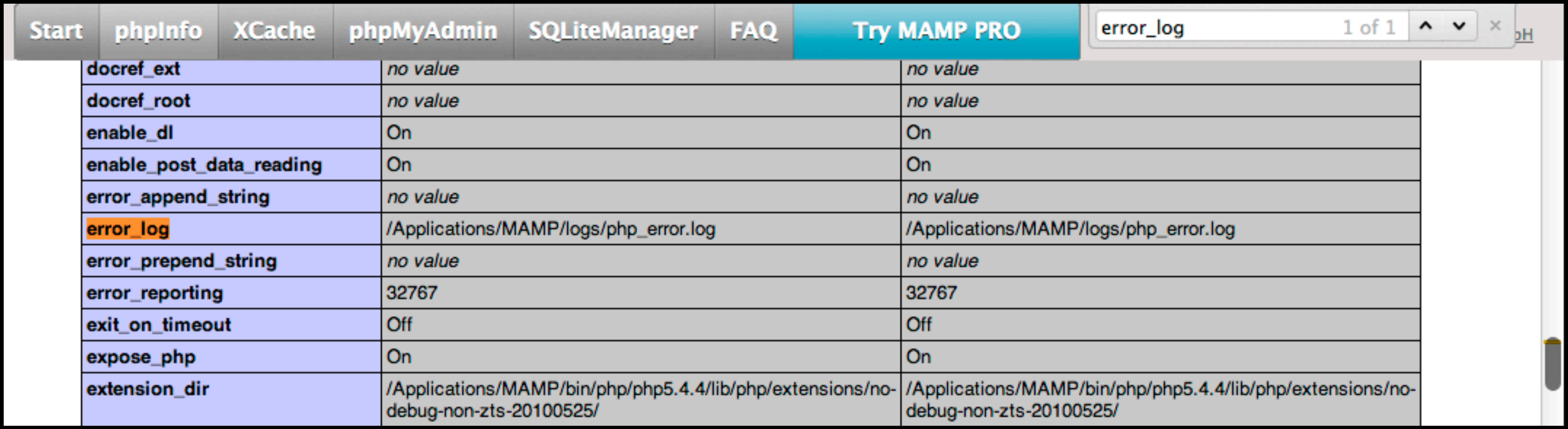
Unable to retrieve record.

Chuck	csev@umich.edu	123	Edit / Delete
Glenn	gg@umich.edu	999	Edit / Delete
Sally	sally@umich.edu	zap	Edit / Delete

Add New

Where do error_log()'s go?

- When in doubt look at PHPInfo

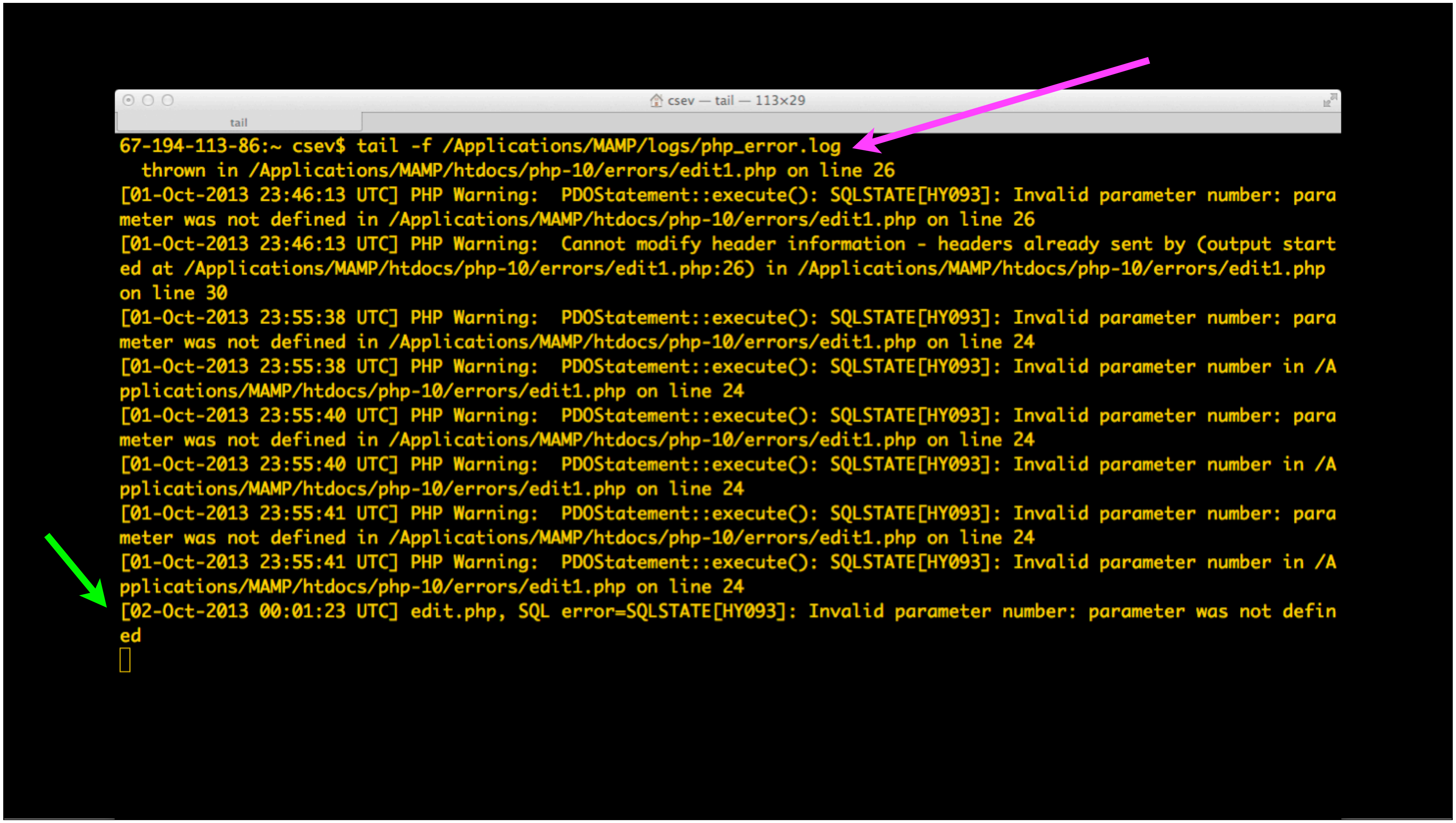


docref_ext	*no value*	*no value*
docref_root	*no value*	*no value*
enable_dl	On	On
enable_post_data_reading	On	On
error_append_string	*no value*	*no value*
error_log	/Applications/MAMP/logs/php_error.log	/Applications/MAMP/logs/php_error.log
error_prepend_string	*no value*	*no value*
error_reporting	32767	32767
exit_on_timeout	Off	Off
expose_php	On	On
extension_dir	/Applications/MAMP/bin/php/php5.4.4/lib/php/extensions/no-debug-non-zts-20100525/	/Applications/MAMP/bin/php/php5.4.4/lib/php/extensions/no-debug-non-zts-20100525/

Where do error_log()'s go?

- /Applications/MAMP/logs/php_error.log
- /Applications/XAMPP/logs/php_error.log
- c:\xampp\php\logs\php_error_log

- On Mac / Linux use 'tail -f filename'



```
67-194-113-86:~ csev$ tail -f /Applications/MAMP/logs/php_error.log
  thrown in /Applications/MAMP/htdocs/php-10/errors/edit1.php on line 26
[01-Oct-2013 23:46:13 UTC] PHP Warning:  PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number: para
meter was not defined in /Applications/MAMP/htdocs/php-10/errors/edit1.php on line 26
[01-Oct-2013 23:46:13 UTC] PHP Warning:  Cannot modify header information - headers already sent by (output start
ed at /Applications/MAMP/htdocs/php-10/errors/edit1.php:26) in /Applications/MAMP/htdocs/php-10/errors/edit1.php
on line 30
[01-Oct-2013 23:55:38 UTC] PHP Warning:  PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number: para
meter was not defined in /Applications/MAMP/htdocs/php-10/errors/edit1.php on line 24
[01-Oct-2013 23:55:38 UTC] PHP Warning:  PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number in /A
pplications/MAMP/htdocs/php-10/errors/edit1.php on line 24
[01-Oct-2013 23:55:40 UTC] PHP Warning:  PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number: para
meter was not defined in /Applications/MAMP/htdocs/php-10/errors/edit1.php on line 24
[01-Oct-2013 23:55:40 UTC] PHP Warning:  PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number in /A
pplications/MAMP/htdocs/php-10/errors/edit1.php on line 24
[01-Oct-2013 23:55:41 UTC] PHP Warning:  PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number: para
meter was not defined in /Applications/MAMP/htdocs/php-10/errors/edit1.php on line 24
[01-Oct-2013 23:55:41 UTC] PHP Warning:  PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number in /A
pplications/MAMP/htdocs/php-10/errors/edit1.php on line 24
[02-Oct-2013 00:01:23 UTC] edit.php, SQL error=SQLSTATE[HY093]: Invalid parameter number: parameter was not defin
ed
```

Summary

- Making database connections
- Doing database operations
- SQL security (a.k.a. we love PDO prepared statements)
- A multi-file CRUD application with redirect
- Exploring errors...