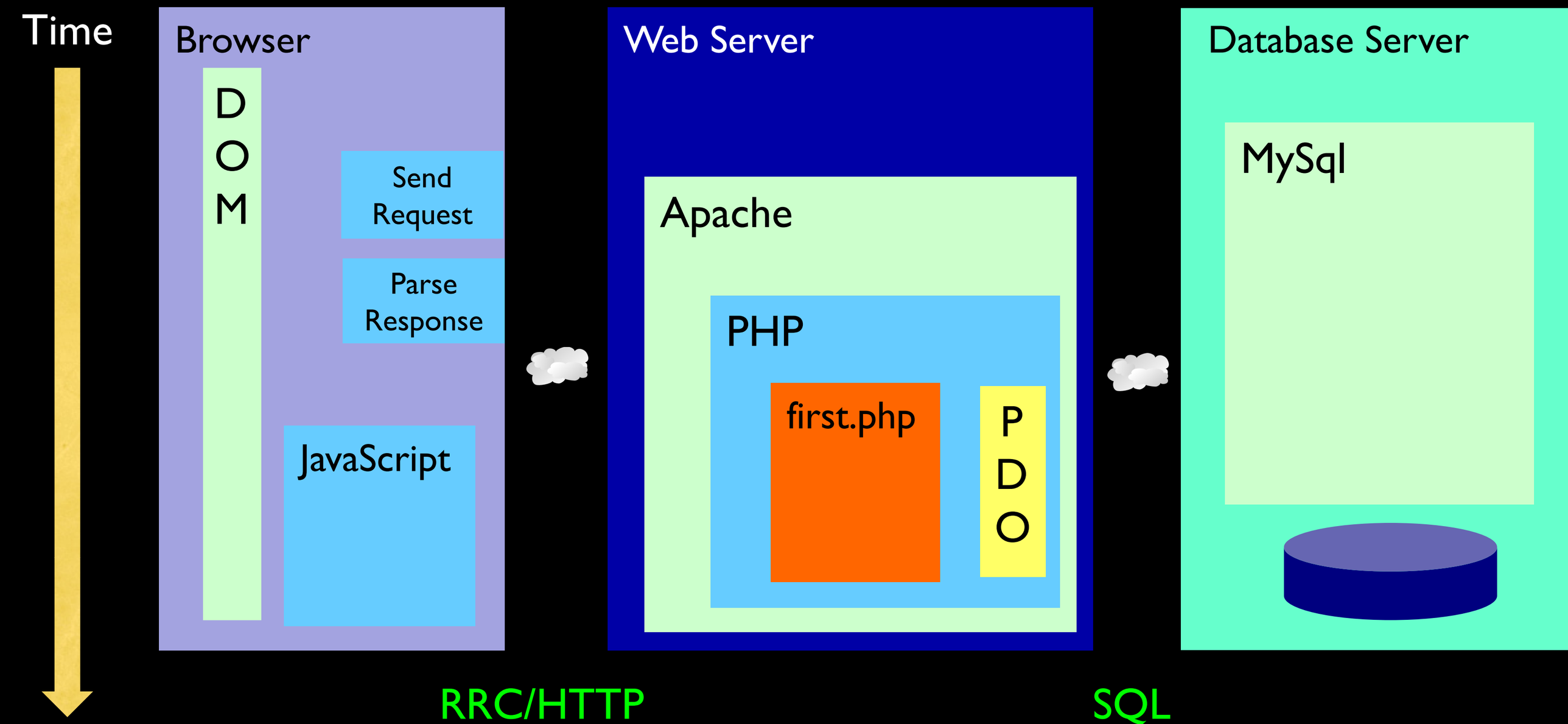
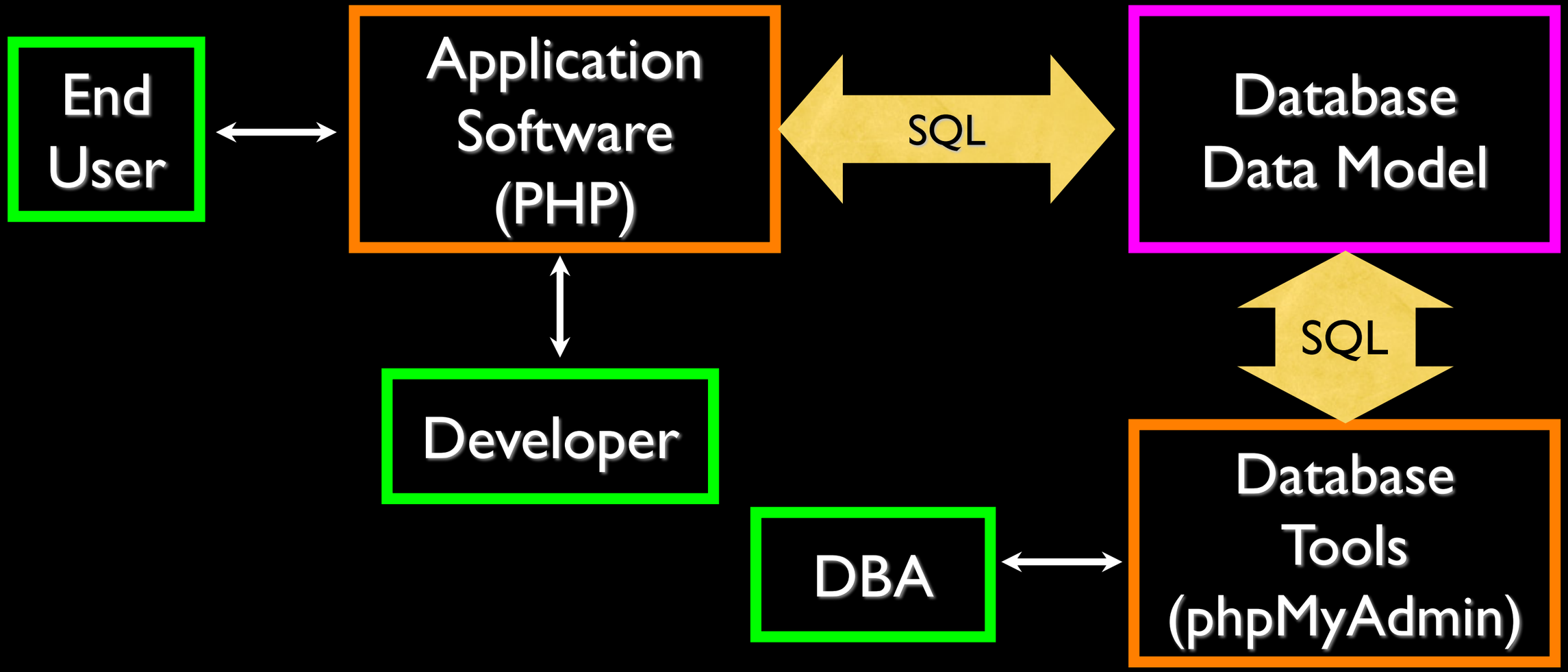
Connecting PHP and SQL

Charles Severance
www.wa4e.com

http://www.wa4e.com/code/pdo.zip


Time
Browser
DOM
Send Request
Parse Response
JavaScript
Web Server
Apache
PHP
first.php
PDO
Database Server
MySql
RRC/HTTP
SQL


End User
Application Software (PHP)
SQL
Database Data Model
SQL
Developer
DBA
Database Tools (phpMyAdmin)

Multiple Ways to Access MySql

- PHP is evolving - there are three ways to access MySql
  - 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

Running SQL Queries in PHP

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 at 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 (
    user_id INTEGER NOT NULL
      AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(128),
    email VARCHAR(128),
    password VARCHAR(128),
    INDEX(email)
) ENGINE=InnoDB CHARSET=utf8;
```

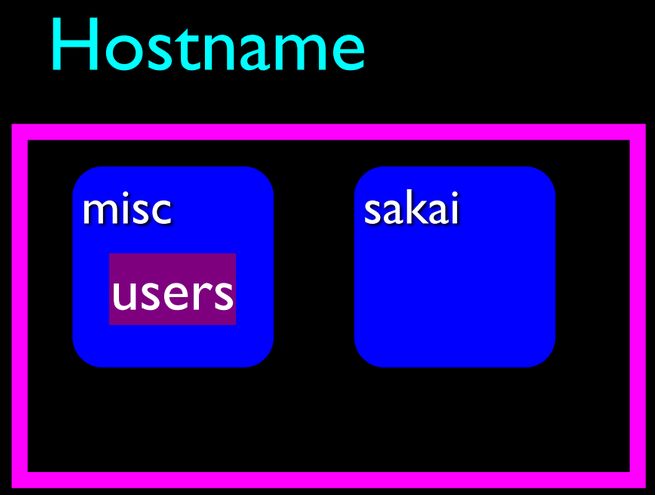
```
mysql> describe users;
+----------+------------------+------+-----+---------+----------------+
| Field    | Type             | Null | Key | Default | Extra          |
+----------+------------------+------+-----+---------+----------------+
| user_id  | int(11)          | NO   | PRI | NULL    | auto_increment |
| 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;
+---------+-------+-----------------+----------+
| user_id | name  | email           | password |
+---------+-------+-----------------+----------+
|       1 | Chuck | csev@umich.edu  | 123      |
|       2 | Glenn | gg@umich.edu    | 456      |
+---------+-------+-----------------+----------+
```

Database Connection



http://www.wa4e.com/code/pdo.zip

pdo.php

Database Connection

3306 for xampp/linux

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

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;
+---------+-------+-----------------+----------+
| user_id | name  | email           | password |
+---------+-------+-----------------+----------+
|       1 | Chuck | csev@umich.edu  | 123      |
|       2 | Glenn | gg@umich.edu    | 456      |
+---------+-------+-----------------+----------+
```

```
Array(
    [user_id] => 1
    [name] => Chuck
    [email] => csev@umich.edu
    [password] => 123
)

Array(
    [user_id] => 2
    [name] => Glenn
    [email] => gg@umich.edu
    [password] => 456
)
```

second.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

```
<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 :)

pdo.php

3306 for xampp/linux

```
<?php
$pdo = new PDO('mysql:host=localhost;port=8889;dbname=misc',
    'fred', 'zap');
// See the "errors" folder for details...
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

third.php

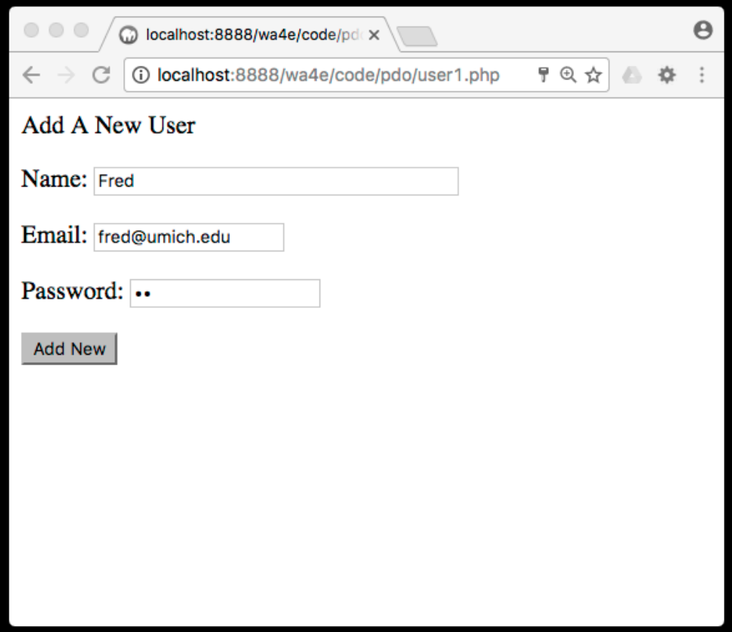
```
<?php
echo "<pre>\n";
require_once "pdo.php";

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

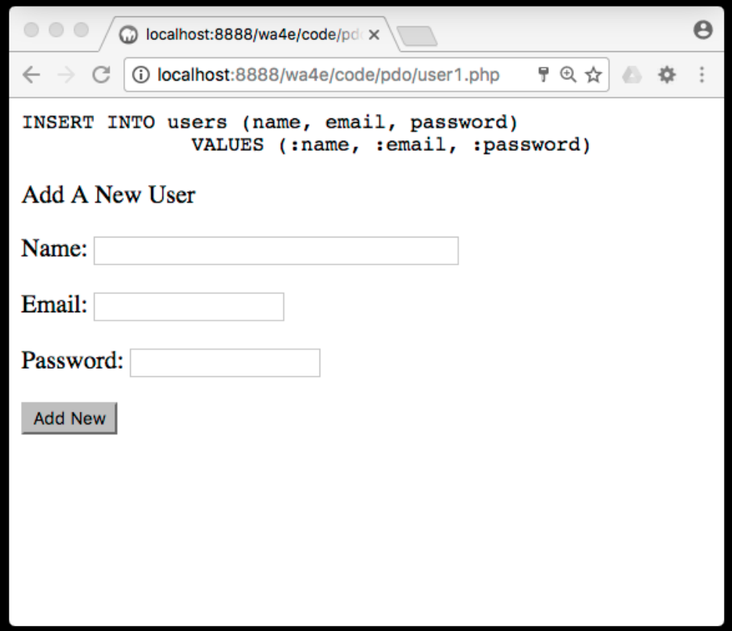
```
Array(
    [user_id] => 1
    [name] => Chuck
    [email] => csev@umich.edu
    [password] => 123
)
Array(
    [user_id] => 2
    [name] => Glenn
    [email] => gg@umich.edu
    [password] => 456
)
```

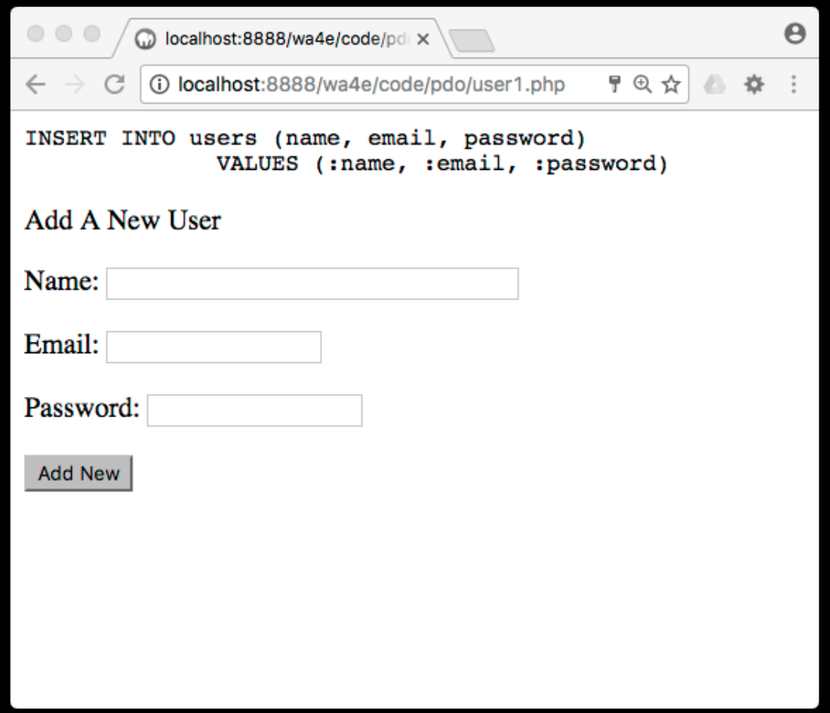
Accesing MySQL Using PDO: Inserting Data

```
<?php
require_once "pdo.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>
```

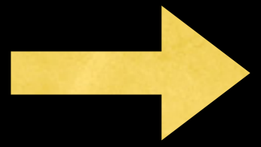
user1.php







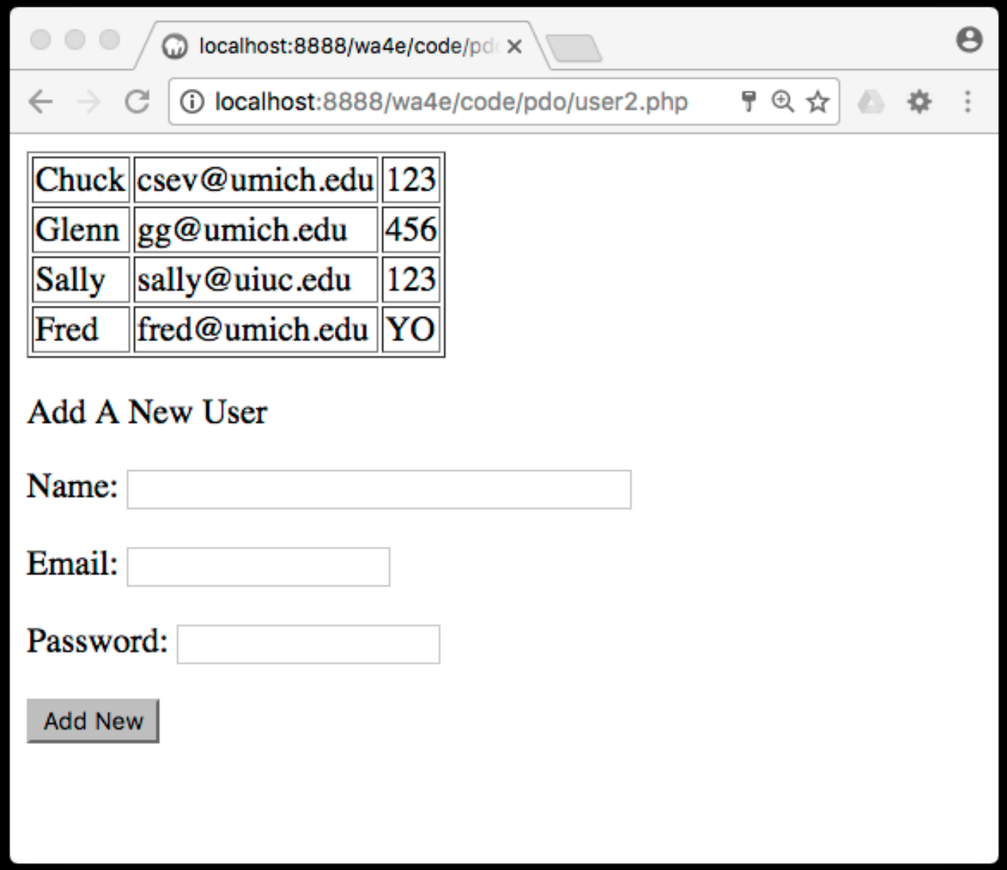
user1.php



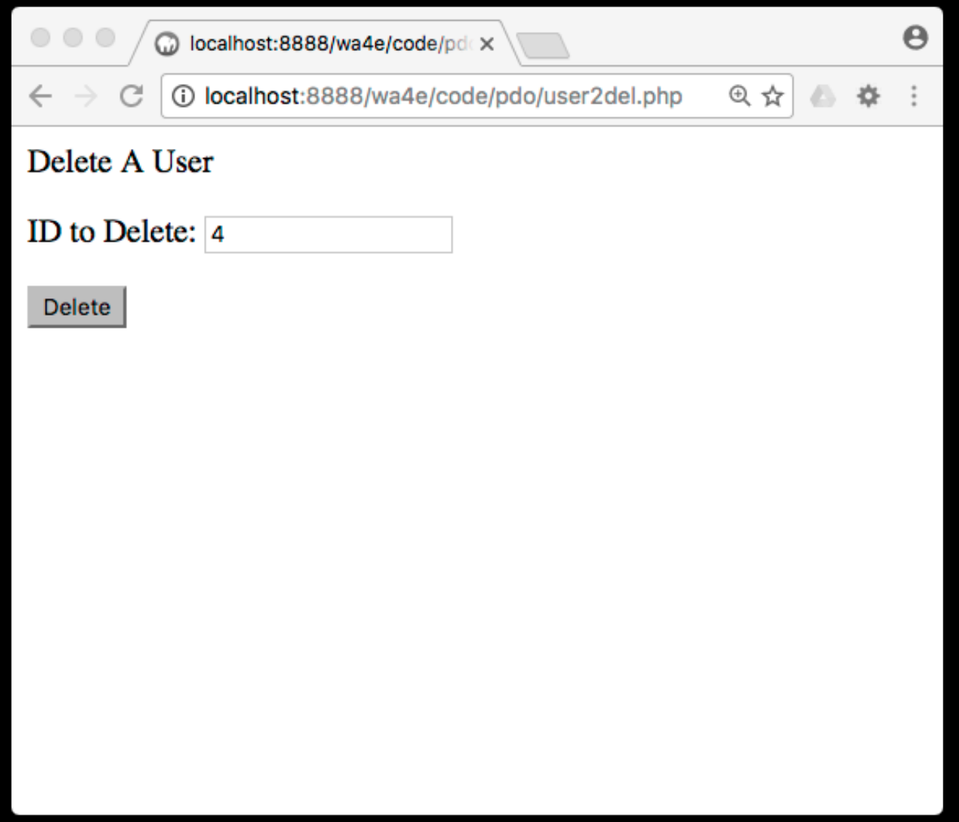
```
mysql> select * from users;
+---------+-------+----------------+----------+
| user_id | name  | email          | password |
+---------+-------+----------------+----------+
|       1 | Chuck | csev@umich.edu | 123      |
|       2 | Glenn | gg@umich.edu   | 456      |
|       3 | Sally | sally@uiuc.edu | 123      |
|       4 | Fred  | fred@umich.edu | YO       |
+---------+-------+----------------+----------+
```

```
 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><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>
```

user2.php

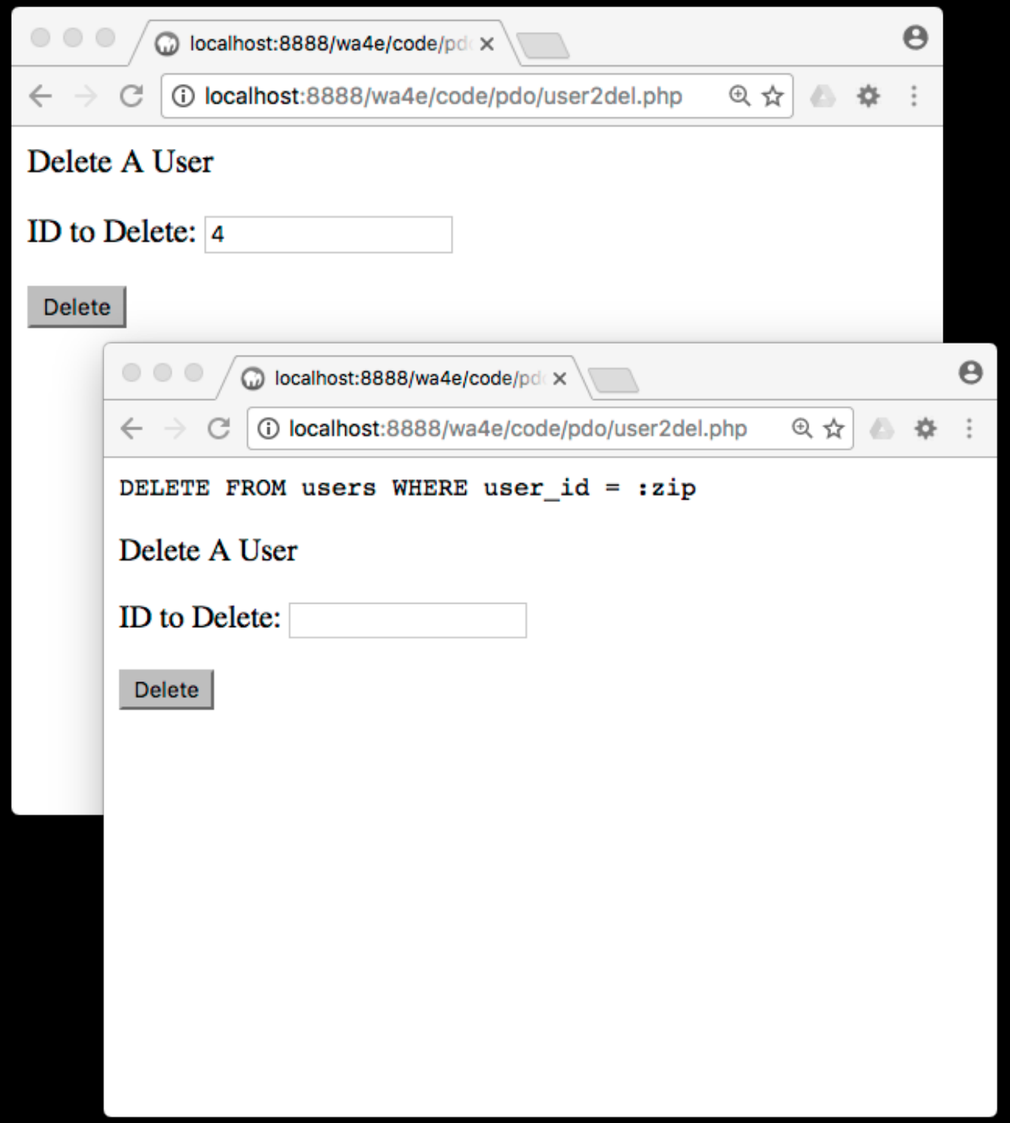


```php
<?php
require_once "pdo.php";

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

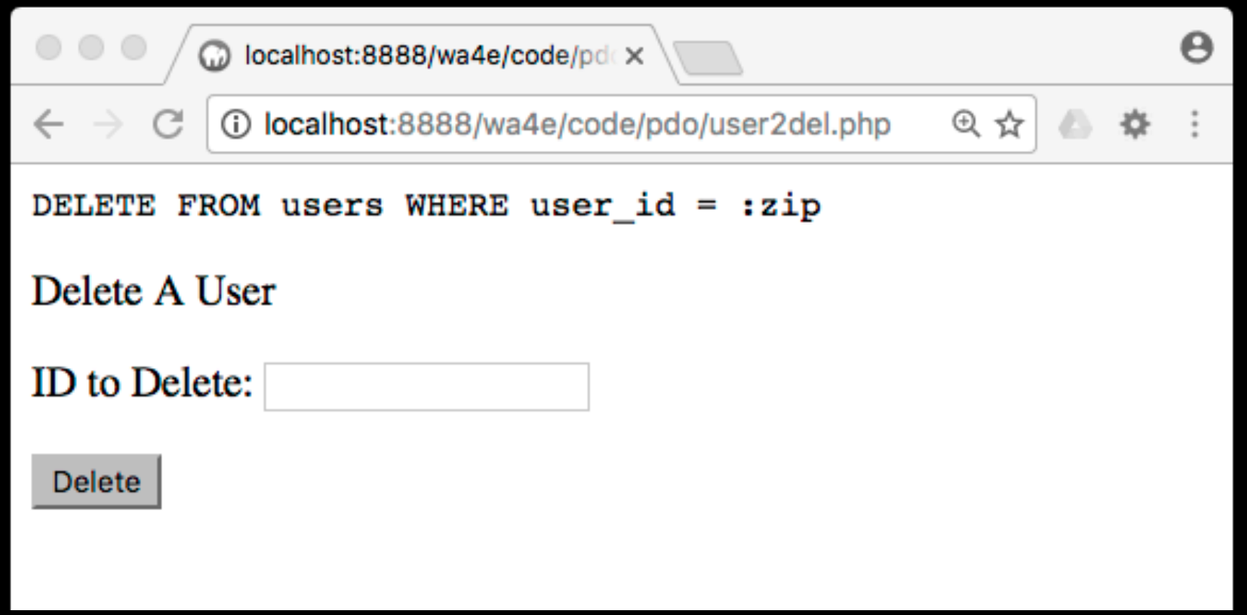


user2del.php

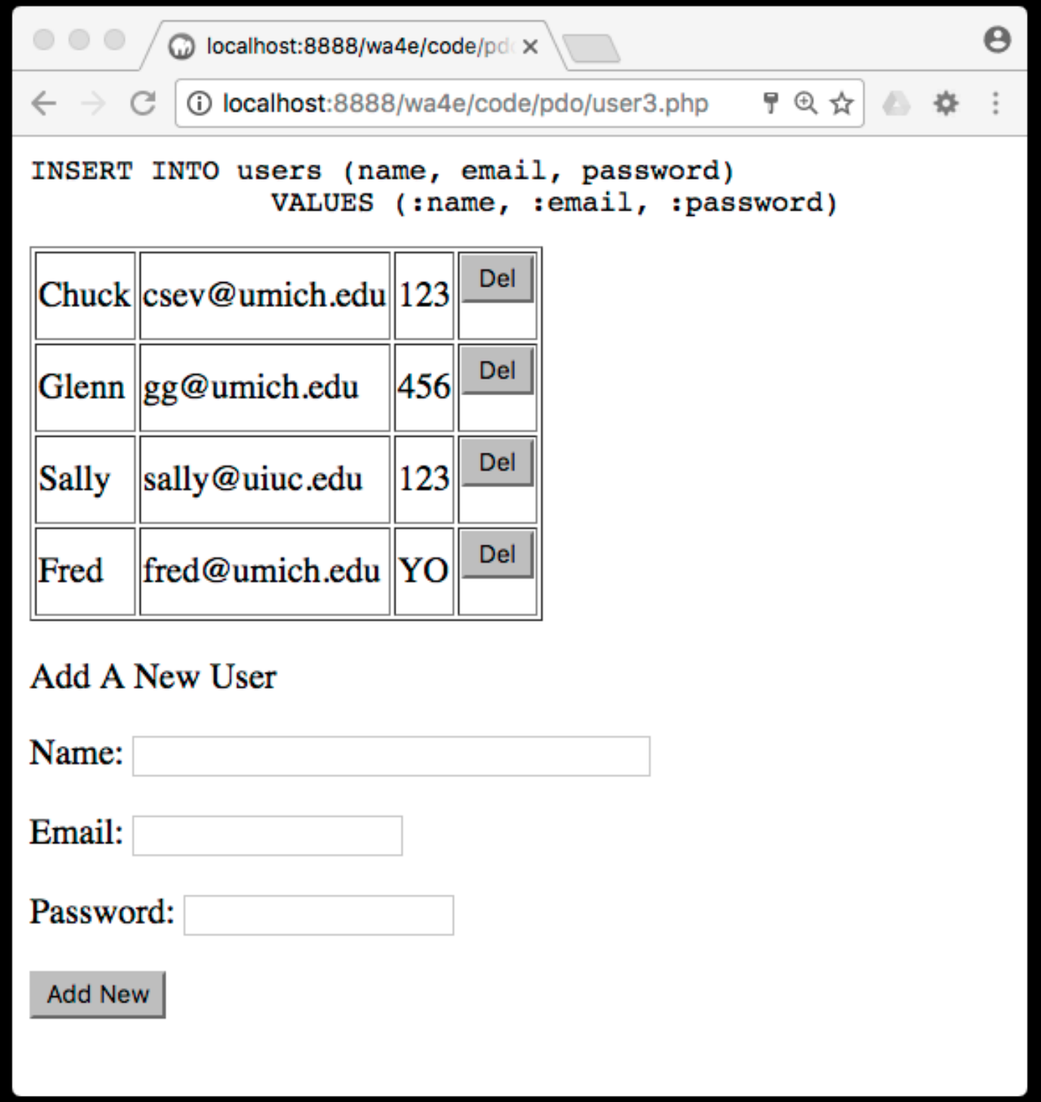
```php
<?php
require_once "pdo.php";

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



user2del.php




```
mysql> select * from users;
+---------+-------+-----------------+----------+
| user_id | name  | email           | password |
+---------+-------+-----------------+----------+
|       1 | Chuck | csev@umich.edu  | 123      |
|       2 | Glenn | gg@umich.edu    | 456      |
|       3 | Sally | sally@uiuc.edu  | 123      |
+---------+-------+-----------------+----------+
```

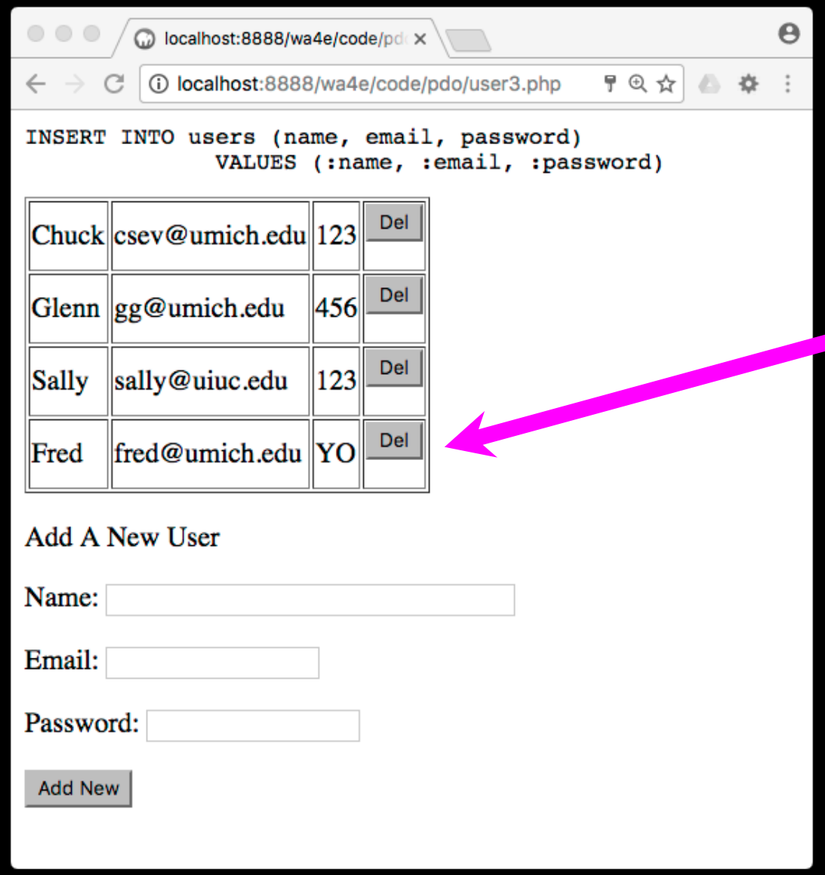

localhost:8888/wa4e/code/pd
localhost:8888/wa4e/code/pdo/user3.php
INSERT INTO users (name, email, password)
VALUES (:name, :email, :password)
Chuck csev@umich.edu 123 Del
Glenn gg@umich.edu 456 Del
Sally sally@uiuc.edu 123 Del
Fred fred@umich.edu YO Del
Add A New User
Name:
Email:
Password:
Add New

user3.php

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

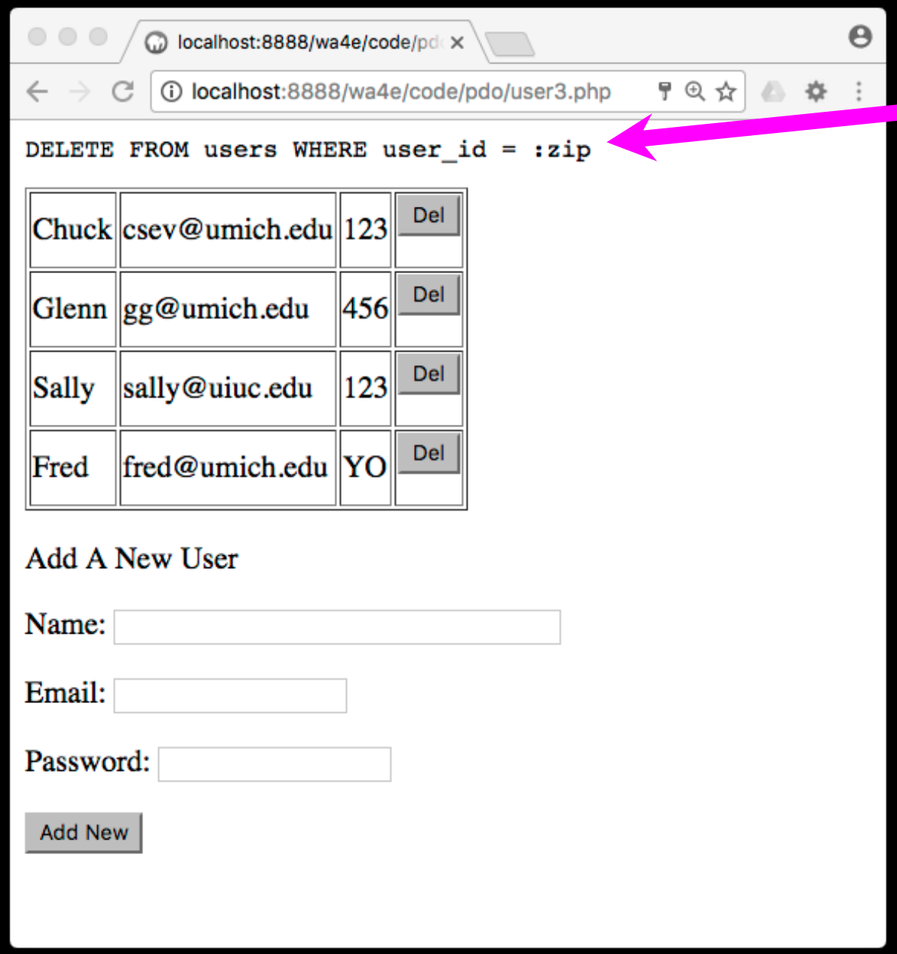
user3.php

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




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

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




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

Program Outline

```
<?php

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

```
<?php
require_once "pdo.php";I
f ( 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['user_id']) ) {
    $sql = "DELETE FROM users WHERE user_id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['user_id']));
}
?>
<html><head></head>
<body>
<table border="1">
<?php
$stmt = $pdo->query("SELECT name, email, password, user_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="huser_idden" ');
    echo('name="user_id" value="'.$row['user_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, user_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="user_id" value="'.$row['user_id'].'">'."\n");
    echo('<input type="submit" value="Del" name="delete">');
    echo("\n</form>\n");
    echo("</td></tr>\n");
}
?>
</table>
```

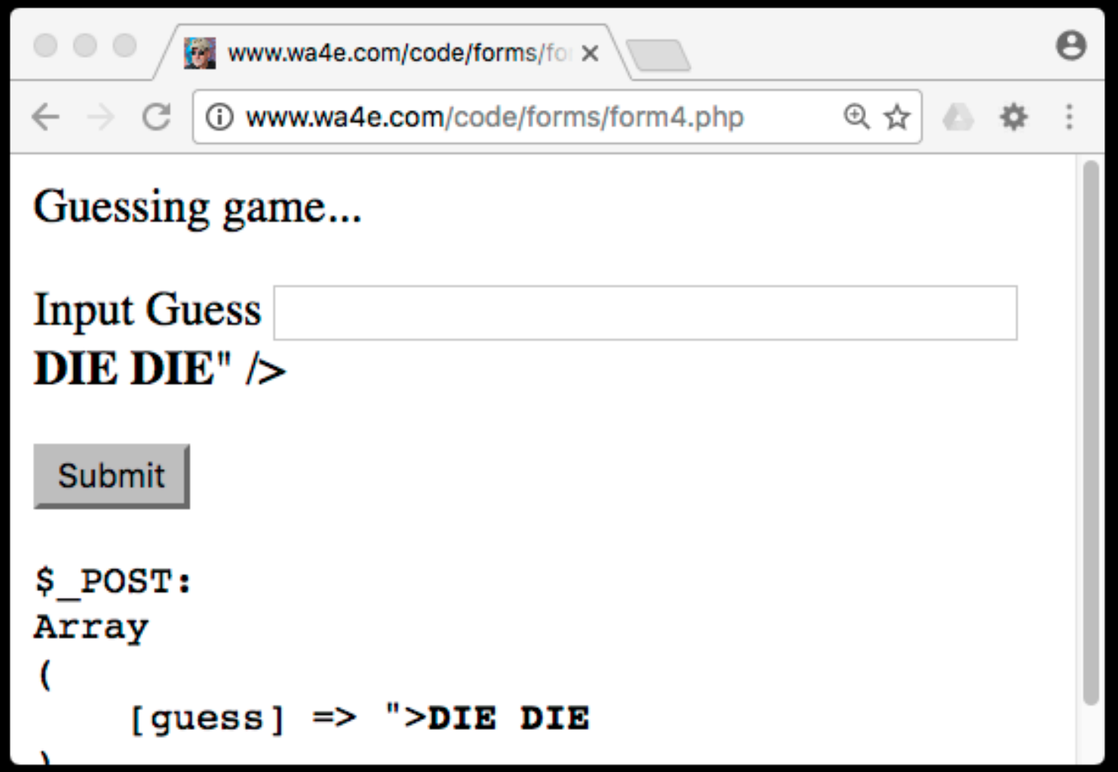
```
<?php
require_once "pdo.php";I
f ( isset($_POST['name']) && isset($_POST['email'])
     && isset($_POST['password'])) {
    $sql = "INSERT INTO users (name, email, password)
            VALUES (:name, :email, :password)";
    echo("<pre>\n".$sql."\n</pre>\n");
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(
        ':name' => $_POST['name'],
        ':email' => $_POST['email'],
        ':password' => $_POST['password']));
}
if ( isset($_POST['delete']) && isset($_POST['user_id']) ) {
    $sql = "DELETE FROM users WHERE user_id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['user_id']));
}
?>
<html><head></head>
<body>
<table border="1">
<?php
$stmt = $pdo->query("SELECT name, email, password, user_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="huser_idden" ');
    echo('name="user_id" value="'.$row['user_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>
```

```
<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 "pdo.php";I
f ( 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['user_id']) ) {
    $sql = "DELETE FROM users WHERE user_id = :zip";
    echo "<pre>\n$sql\n</pre>\n";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(array(':zip' => $_POST['user_id']));
}
?>
<html><head></head>
<body>
<table border="1">
<?php
$stmt = $pdo->query("SELECT name, email, password, user_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="huser_idden" ');
    echo('name="user_id" value="'.$row['user_id'].'">'."\n");
    echo('<input type="submit" value="Del" name="delete">');
    echo("\n</form>\n");
    echo("</td></tr>\n");
}
?>
<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 Issue: Avoiding SQL Injection

Recall HTML Injection ...




```
<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>
```

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

SQL Injection

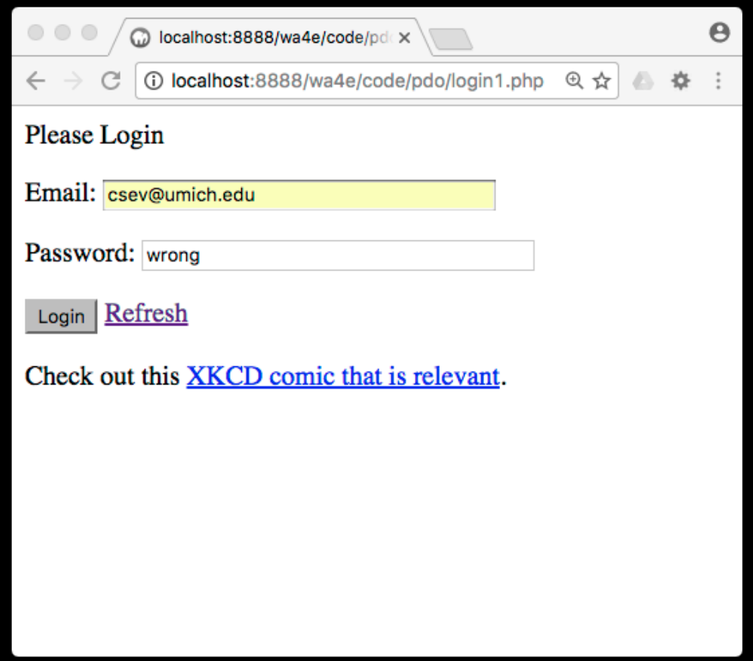
This code does it all in a select instead of a prepare/execute pattern, but it is prone to SQL Injection – where and why?

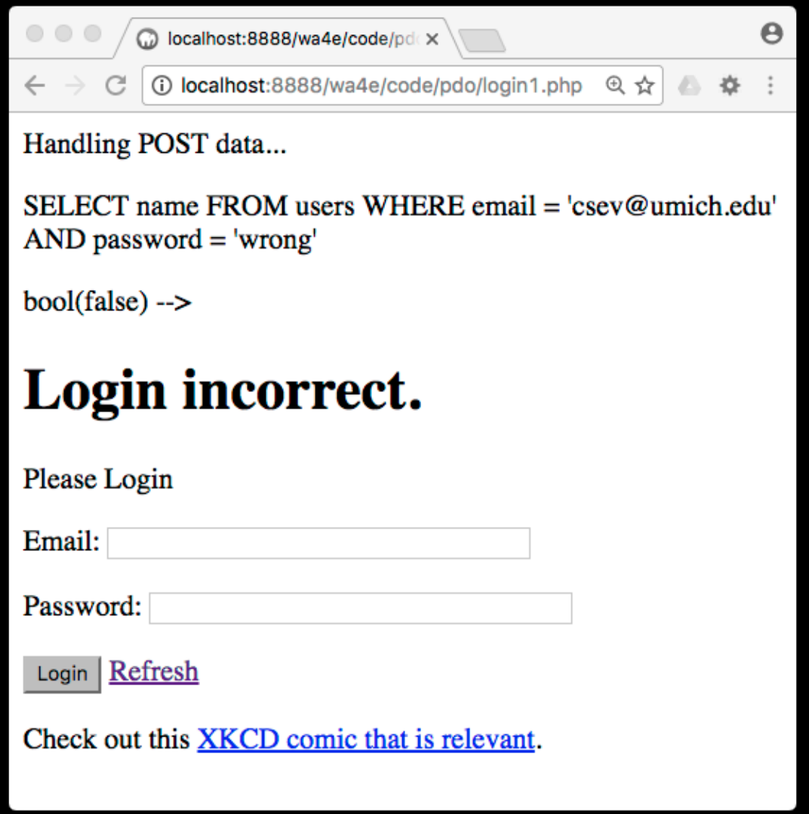
```
if ( isset($_POST['email']) && isset($_POST['password']) ) {
    $e = $_POST['email'];
    $p = $_POST['password'];
    $sql = "SELECT name FROM users
          WHERE email = '$e'
          AND password = '$p'";
    $stmt = $pdo->query($sql);
```

login1.php

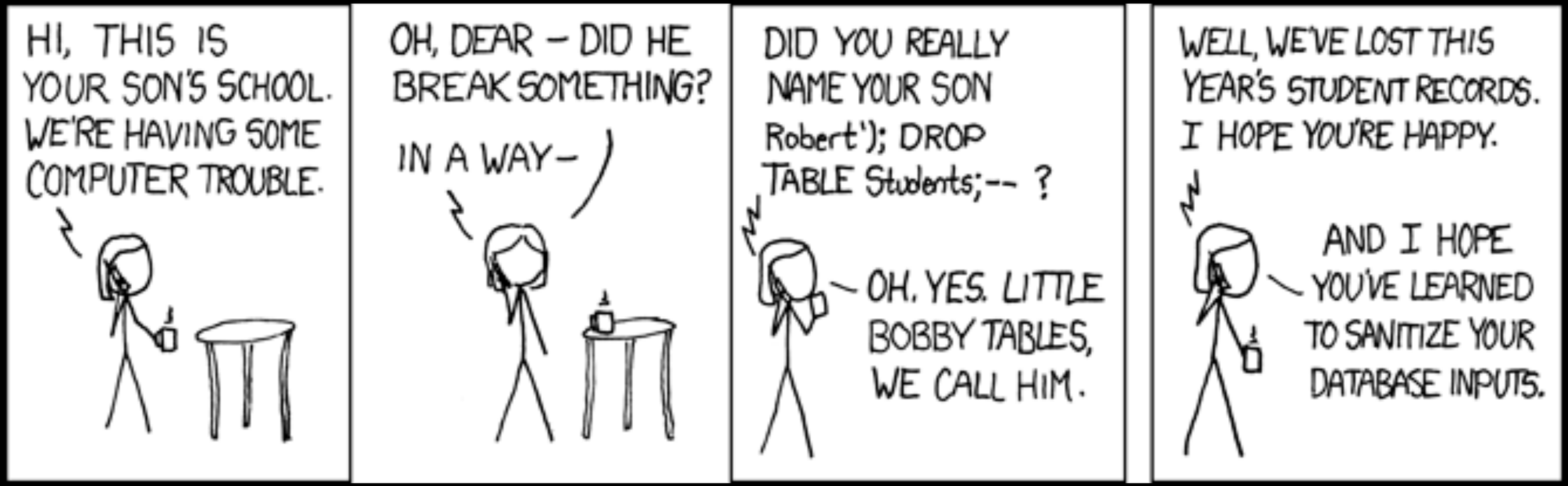
id	name	email	password
1	Chuck	csev@umich.edu	123
4	Sally	sally@umich.edu	zap
13	Sarah	sarah@umich.edu	123
17	Barb	barb@umich.edu	123
19	Bill	bill@umich.edu	123

What Could Go Wrong?



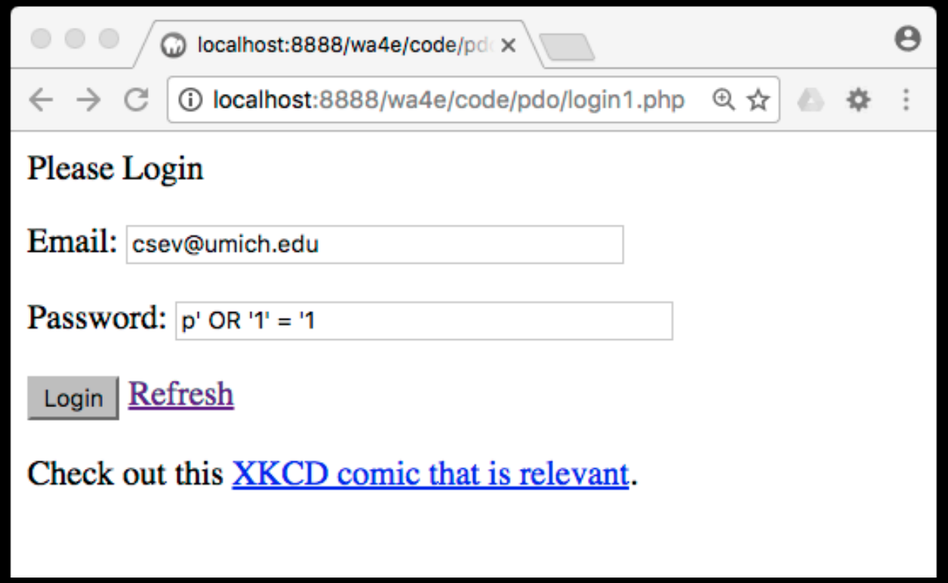


login1.php



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'";
    $stmt = $pdo->query($sql);
```



login1.php

Use Prepared Statements Properly

```
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);
```

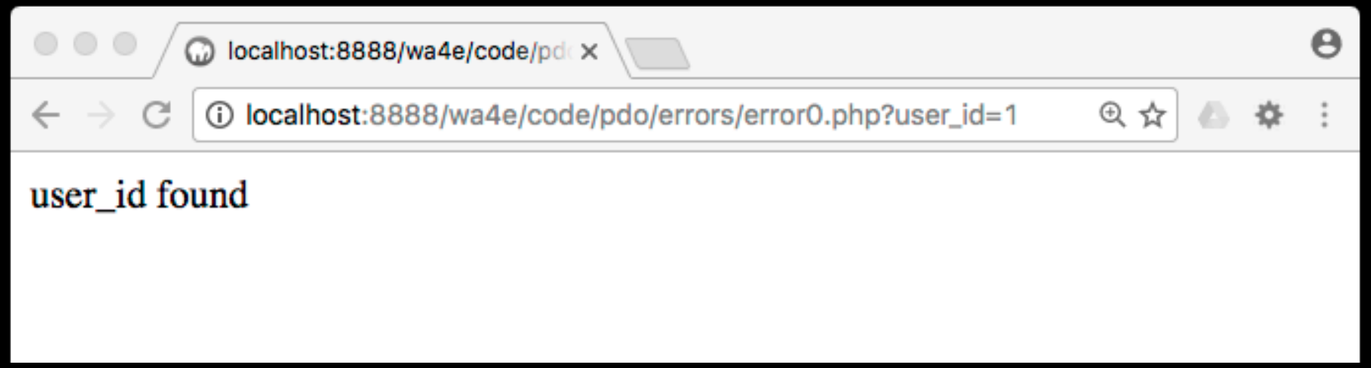
login2.php

When the statement is executed, the placeholders get replaced with the actual strings and everything is automatically escaped!

Error Handling with PDO

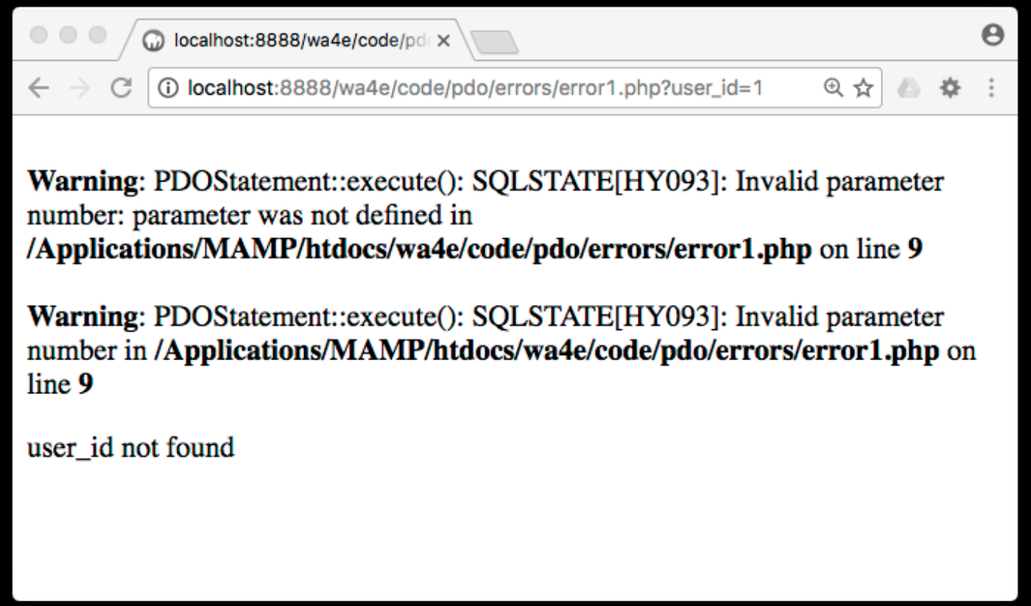
```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);

$stmt = $pdo->prepare("SELECT * FROM users where user_id = :xyz");
$stmt->execute(array(":xyz" => $_GET['user_id']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
if ( $row === false ) {
    echo("<p>user_id not found</p>\n");
} else {
    echo("<p>user_id found</p>\n");
}
```

errors/error0.php



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

$stmt = $pdo->prepare("SELECT * FROM users where user_id = :xyz");
$stmt->execute(array(":pizza" => $_GET['user_id']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
if ( $row === false ) {
    echo("<p>user_id not found</p>\n");
} else {
    echo("<p>user_id found</p>\n");
}
```

errors/error0.php



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`

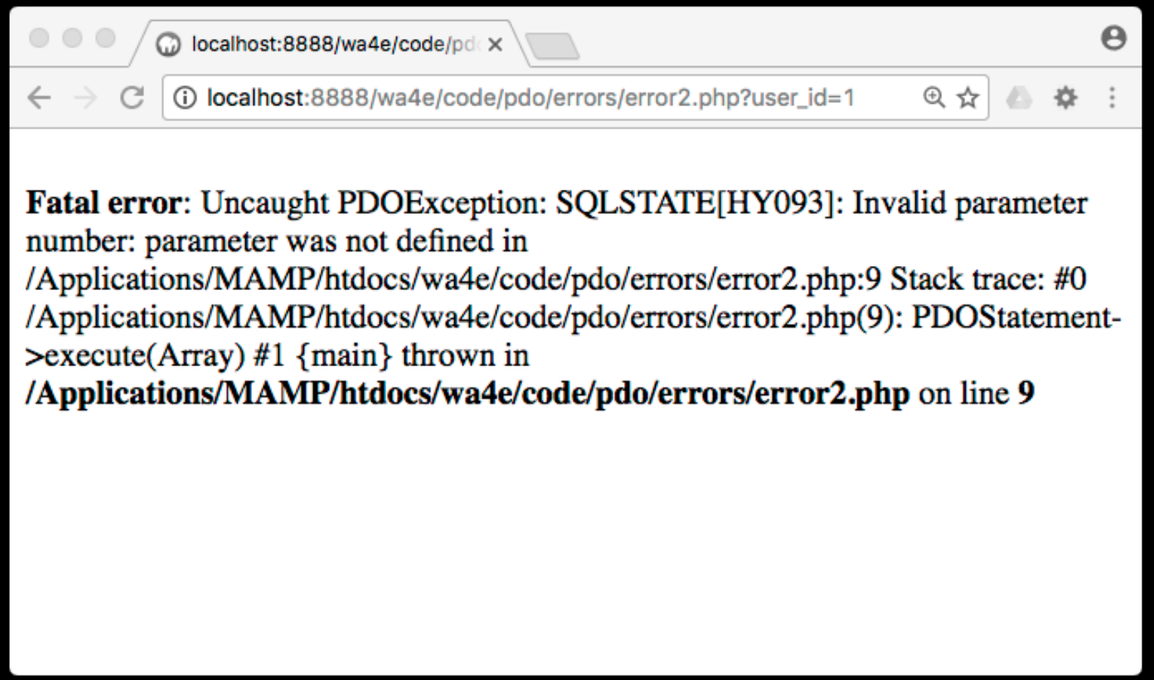
  In addition to setting the error code, PDO will throw a PDOException and set its properties to reflect the error code and error information. This setting is also useful during debugging, as it will effectively "blow up" the script at the point of the error, very quickly pointing a finger at potential 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_EXCEPTION);
$stmt = $pdo->prepare("SELECT * FROM users where user_id = :xyz");
$stmt->execute(array(":pizza" => $_GET['user_id']));
$row = $stmt->fetch(PDO::FETCH_ASSOC);
if ( $row === false ) {
    echo("<p>user_id not found</p>\n");
} else {
    echo("<p>user_id found</p>\n");
}
```



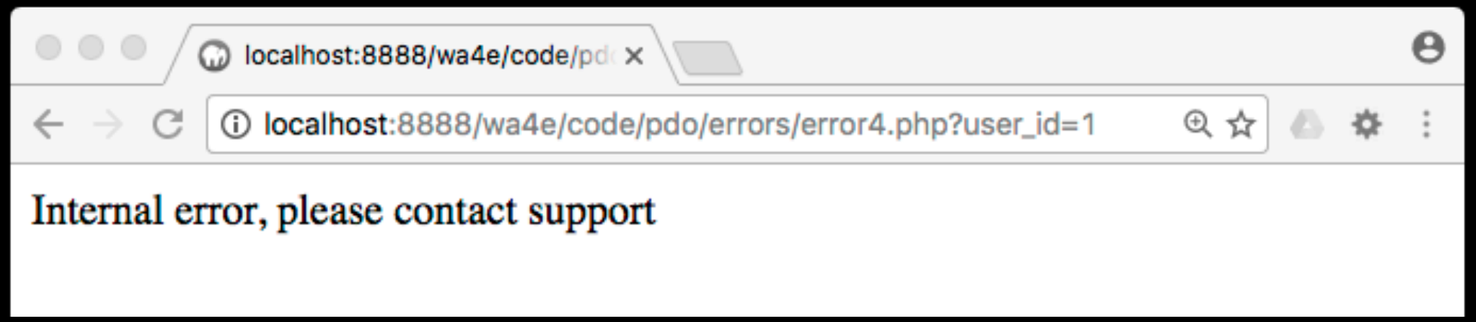
errors/error2.php

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
try {
    $stmt = $pdo->prepare("SELECT * FROM users where user_id = :xyz");
    $stmt->execute(array(":pizza" => $_GET['user_id']));
} catch (Exception $ex ) {
    echo("Exception message: ".$ex->getMessage());
    return;
}
$row = $stmt->fetch(PDO::FETCH_ASSOC);
```

errors/error3.php

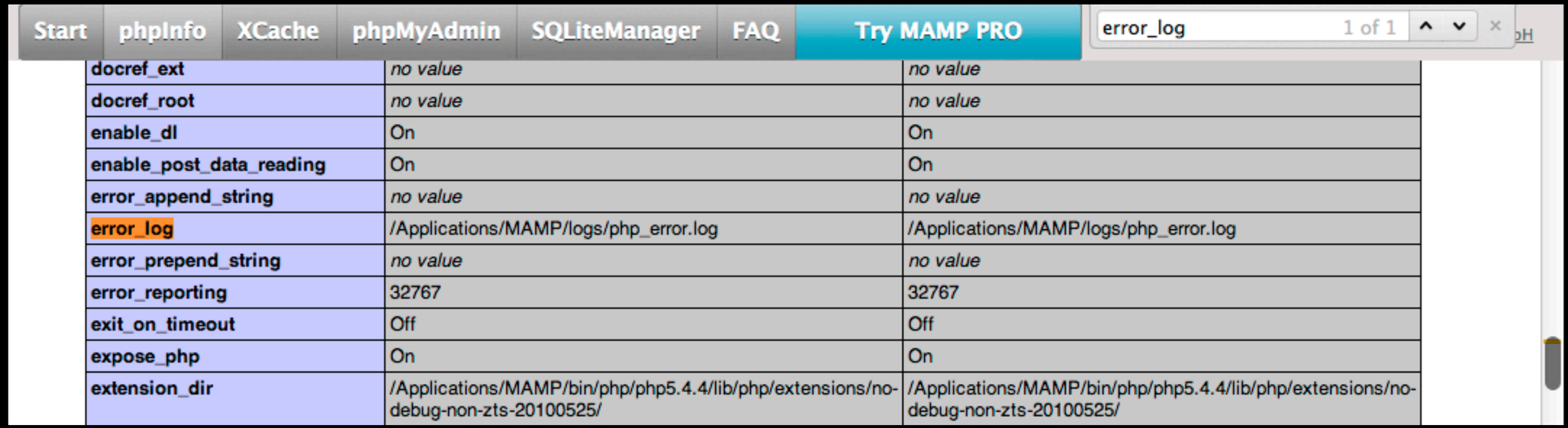
```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
try {
    $stmt = $pdo->prepare("SELECT * FROM users where user_id = :xyz");
    $stmt->execute(array(":pizza" => $_GET['user_id']));
} catch (Exception $ex ) {
    echo("Internal error, please contact support");
    error_log("error4.php, SQL error=".$ex->getMessage());
    return;
}
$row = $stmt->fetch(PDO::FETCH_ASSOC);
```

errors/error4.php



Where do error_log()s go?

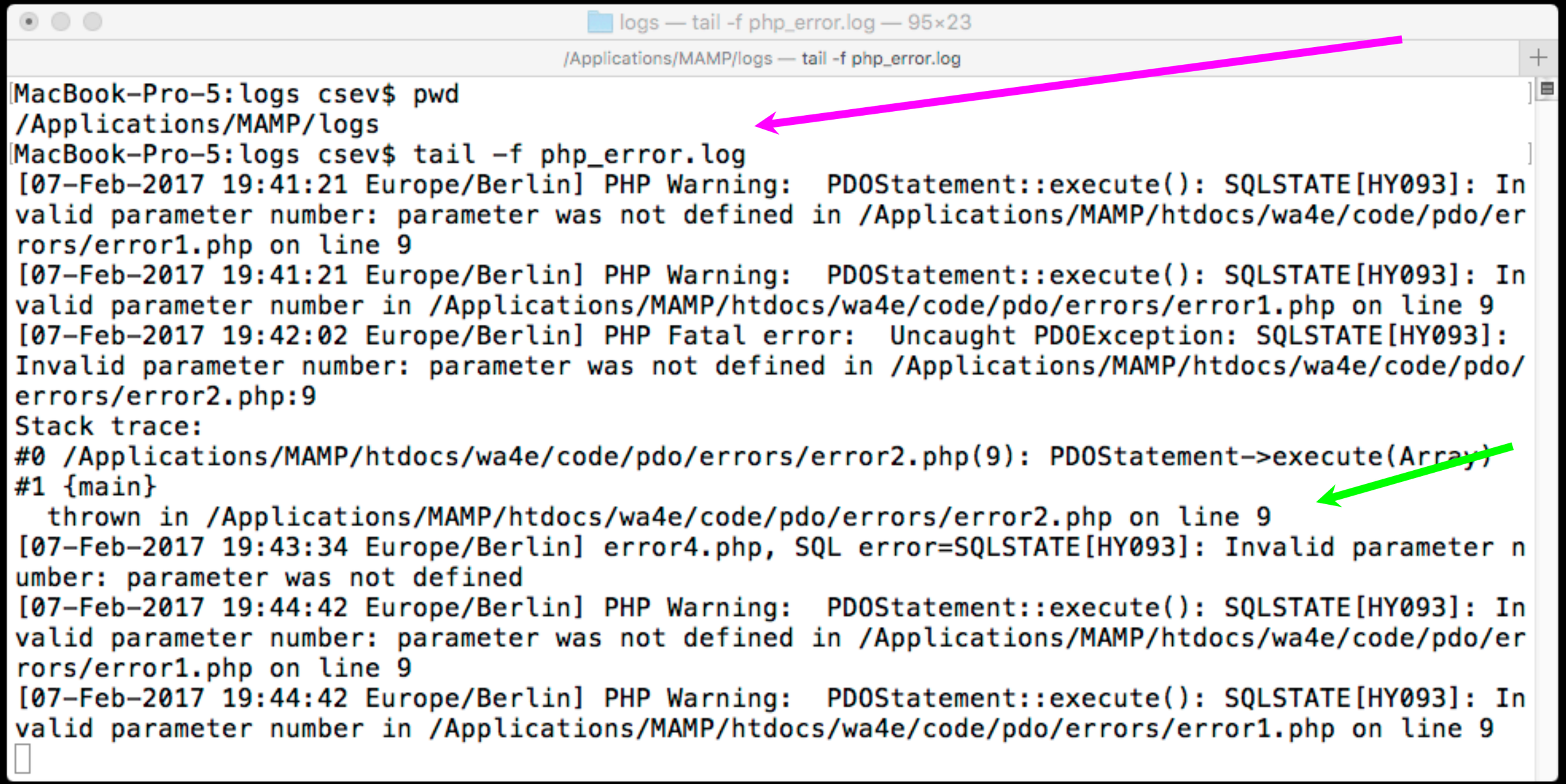
When in doubt, look at PHPInfo...



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?

- File Paths:
  - /Applications/MAMP/logs/php_error.log
  - c:\xampp\php\logs\php_error_log
- Open the log file and scroll to the bottom
- Watch the log actively
  - On Mac / Linux use: tail -f filename
  - Windows: http://ophilipp.free.fr/op_tail.htm


logs — tail -f php_error.log — 95×23
/Applications/MAMP/logs — tail -f php_error.log
MacBook-Pro-5:logs csev$ pwd
/Applications/MAMP/logs
MacBook-Pro-5:logs csev$ tail -f php_error.log
[07-Feb-2017 19:41:21 Europe/Berlin] PHP Warning: PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number: parameter was not defined in /Applications/MAMP/htdocs/wa4e/code/pdo/errors/error1.php on line 9
[07-Feb-2017 19:41:21 Europe/Berlin] PHP Warning: PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number in /Applications/MAMP/htdocs/wa4e/code/pdo/errors/error1.php on line 9
[07-Feb-2017 19:42:02 Europe/Berlin] PHP Fatal error: Uncaught PDOException: SQLSTATE[HY093]: Invalid parameter number: parameter was not defined in /Applications/MAMP/htdocs/wa4e/code/pdo/errors/error2.php:9
Stack trace:
#0 /Applications/MAMP/htdocs/wa4e/code/pdo/errors/error2.php(9): PDOStatement->execute(Array)
#1 {main}
thrown in /Applications/MAMP/htdocs/wa4e/code/pdo/errors/error2.php on line 9
[07-Feb-2017 19:43:34 Europe/Berlin] error4.php, SQL error=SQLSTATE[HY093]: Invalid parameter number: parameter was not defined
[07-Feb-2017 19:44:42 Europe/Berlin] PHP Warning: PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number: parameter was not defined in /Applications/MAMP/htdocs/wa4e/code/pdo/errors/error1.php on line 9
[07-Feb-2017 19:44:42 Europe/Berlin] PHP Warning: PDOStatement::execute(): SQLSTATE[HY093]: Invalid parameter number in /Applications/MAMP/htdocs/wa4e/code/pdo/errors/error1.php on line 9

Summary

- Making database connections
- Doing database operations
- SQL security (a.k.a. we love PDO prepared statements)
- Exploring errors...

Acknowledgements / Contributions

Initial Development: Charles Severance, University of Michigan School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators here