

PHP and MVC Architecture

Build a better website

Class 2: Controllers and Data Manipulation

Getting data in and out



PHP and HTML Forms

HTML Forms -- views/blogger/list.php

New Blogger Form

```
<form action="<?php echo htmlentities($_SERVER['PHP_SELF']); ?>"
method="post">
    <label>Username</label><input type="text" name="username"/>
    <label>Email</label><input type="text" name="email"/>
    <label>Password</label><input type="password" name="password"/>
    <label>Confirm Password</label><input type="password" name
="confirm_password"/>
    <input type="hidden" value="true" name="create_blogger"/>
    <button type="submit" value="Submit" class="add">
        
        <div class="button-text"> Submit</div>
    </button>
</form>
```



PHP and HTML Forms

HTML Forms -- views/bloggers/list.php

Edit Blogger Form

```
<form action="<?php echo htmlentities($_SERVER['PHP_SELF']); ?>"
method="post">
    <label>Username</label><input type="text" name="username" value="<?
php echo $blogger['username'] ?>">
    <label>Email</label><input type="text" name="email" value="<?php
echo $blogger['email'] ?>">
    <label>Old Password</label><input type="password" name
="old_password"/>
    <label>New Password</label><input type="password" name="password"/>
    <input type="hidden" value="<?php echo $blogger['id'] ?>" name
="id"/>
    <input type="hidden" value="true" name="update_blogger"/>
    <button type="submit" value="Submit" class="edit">
        
        <div class="button-text"> Submit</div>
    </button>
</form>
```



PHP and HTML Forms

HTML Forms -- views/bloggers/list.php

Delete Bloggers Form

```
<form action="<?php echo htmlentities($_SERVER['PHP_SELF']); ?>"
method="post">
    <input type = "hidden" value ="true" name="delete_blogger"/>
    <input type = "hidden" value =<?php echo $blogger['id']?> name ="id"/>
    <button type = "submit" value="Submit" class="delete">
        <img class="icon" src='/public/images/delete.png'/>
        <div class="button-text"> Delete</div>
    </button>
</form>
```



Controllers

Read data -- controllers/post.php

```
<?php
```

```
//initialize data from controller
```

```
$data = Post_Controller::_list();
```

```
?>
```

```
<?php
```

```
class Post_Controller {
```

```
    public static function _list() {
```

```
        $posts_array = Post::getAll();
```

```
        if($posts_array) {
```

```
            foreach($posts_array as $post) {
```

```
                $blogger = Blogger::getOne($post['user_id']);
```

```
                $post['username'] = $blogger['username'];
```

```
            }
```

```
        }
```

```
        return array('posts' => $posts_array);
```

```
    }
```

```
}
```

```
?>
```



Controllers

Create data -- controllers/post.php

```
<?php
class Post_Controller {
    public static function _list() {
        if (isset($_POST['create_post'])) {
            Post::create($_POST);
        }
        //... (code seen on previous slide here)
    }
}
?>
```



Controllers

Update data -- controllers/post.php

```
<?php
class Post_Controller {
    public static function _list() {
        if (isset($_POST['update_post'])) {
            Post::edit($_POST, $_POST['id']);
        }
        //... (code seen on previous slides here)
    }
}
?>
```



Controllers

Delete data -- controllers/post.php

```
<?php
class Post_Controller {
    public static function _list() {
        if (isset($_POST['delete_post'])) {
            Post::destroy($_POST['id']);
        }
        //... (code seen on previous slides here)
    }
}
?>
```



Controllers

Validating data -- controllers/post.php

```
<?php $data = Blogger_Controller::_list();?>
<?php if($data['bloggerWarning']!="") echo $data['bloggerWarning'];?>
```

```
<?php
    class Blogger_Controller {
        public static function _list() {
            $warning = "";
            $blogger_array = Blogger::getAll();
            return array('bloggers' => $blogger_array,
                        'bloggerWarning' => $warning)
        }
    }
?>
```



Controllers

Validating data -- controllers/post.php

```
<?php
class Blogger_Controller {
    public static function _list() {
        $warning = "";
        if (isset($_POST['create_blogger'])) {
            if($_POST['username'] != "" && $_POST['email'] != "" &&
$_POST['password'] != "" && $_POST['confirm_password'] != "") {
                if($_POST['password'] == $_POST['confirm_password']) {
                    Blogger::create($_POST);
                }
            }
            else{
                $warning = 'Password and confirm password must match';
            }
        }
        else{
            $warning = 'Please enter all fields to register a user';
        }
    }
    //..code from previous slide here
}
?>
```



Controllers

Validating data -- controllers/post.php

```
<?php
class Blogger_Controller {
    public static function _list() {
        $warning = "";
        if($_POST['password']== "") {
            Blogger::edit($_POST, $_POST['id']);
        }
        else if($_POST['password'] != "" && $_POST['old_password'] != "") {
            $blogger = Blogger::getOne($_POST['id']);
            if($_POST['old_password'] == $blogger['password']) {
                Blogger::edit($_POST, $_POST['id']);
            }
            else{
                $warning = 'Old password does not match password in the
database';
            }
        }
        else{
            $warning = 'You must provide an old password to change
passwords.';
        }
    }
}
```

//..code from previous slides here



Coming Up!

Securing your website

How to scramble passwords stored into a database

How to allow users to login and logout

How to only allow logged in users to perform certain actions (creating, updating, deleting)



A large, stylized red question mark graphic that serves as a background for the text.

Questions?

