Lesson Number 2

Name:

Connecting / Saving Data

Description:

Using PHP to get input and save input to a database.

GROUP ACTIVITY - PHP Quick Review

5 minutes with a partner to jot down (without looking at notes) what you remember from last week.

ACTIVITY - Exploring Forms

Forms Explained

Purpose

- Why do we use forms?
- How can a form help us to customize a user's experience?

Elements

- What are some common form elements?
    - What element is best used to capture a first or last name, or an email or snail mail address?
    - What element is best used to capture a comment?
    - What element is best used to capture a country or city of origin?
    - What element is best used to capture a choice like a yes or no decision?
    - What element is best used to capture multiple options like a list of sports or movies a person may enjoy?

Explaining METHOD

- the **method** attribute of the form tag allows us to define what type of HTTP request we're making
    - the **POST method** is used for creating new resources within our application
    - the **GET method** is used to retrieve existing resources within our application
    - the **PUT method** is used to update or replace an existing resource within our application
    - the **DELETE method** is used to delete an existing resource within our application
- the **PUT** and **DELETE** methods are unsupported in most browsers
    - Both **PUT** and **DELETE** can be peformed using **POST**
    - Some developers will use a hidden field in their form that passes **PUT** and **DELETE** to the server for processing
    - Many MVC and RESTful applications implement **PUT** and **DELETE** as they are semantically correct and technically idempotent where as **POST** is not

Explaining ACTION

- **action** is the location of the script that will process the form for the user
    - this can be a separate file, the current file, or an external file (off-site)

Explaining ENCTYPE

- **enctype** is the encoding type choosen for the form
    - the default, application/x-www-form-urlencoded
    - multipart/form-data (for when you're uploading files)

Explaining $_POST

- When a form is submitted, and it's method is set to POST, an HTTP Request is created with a body containing the parameters and their values
- PHP processes the request's body into an associative array known as $_POST. $_POST is a super global, meaning it is accessible through all files of your PHP application
- When the form is encoded, it uses the name attribute of each HTML element as a key label in the $_POST associative array
- - For example, $_POST['email'] references the <input name="email"> element
- When a checkbox field is used, PHP will will build a nested array and store it under its key

Example Form
http://comp1006-s16.azurewebsites.net/lesson-02/examples/forms-explored.php

- What information will be stored in the database?
- What information is likely being drawn from existing data in the database?

Saving Data

- What are two places we can store information received from the form?
- What database are WE using in this course to store information?
- What scripting language are we using to process the information?

Important Rules Regarding Our Application's Data

When we build our applications, we want to maintain a few rules in database design:

1. we want to ensure rows in our table are not storing duplicate data (within reason)
2. we want to isolate data that can be divided, like full names, addresses, ingredients, instructions/processes, phone numbers (area code from base number)
3. we want to give our data a unique index so it is distinguishable from other data
4. we want to ensure we are not storing similar data in different fields, such as a table that has the columns artist, singer, actor (these could be condensed into one field, performer)

ACTIVITY - Building our example form

Example File:

/lesson-02/examples/new_artist.php

NOTES:

- Attaching JQuery, Bootstrap, and FontAwesome: https://www.bootstrapcdn.com/

```
<link
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.
" rel="stylesheet" integrity="sha384-
1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7"
crossorigin="anonymous">
```

```
<script src="https://code.jquery.com/jquery-2.2.3.js"
integrity="sha256-laXWtGydpwqJ8JA+X9x2miwmaiKhn8tVmOVEigRNtP4="
crossorigin="anonymous"></script>
```

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.mi
" integrity="sha384-
0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxss1yVqOtnepnHVP9aJ7xS"
crossorigin="anonymous"></script>
```

- Place the CSS in the head and the JavaScript just before the closing </body> tag

- Using HTML5 Doctype

```
<!DOCTYPE html>
```

- We'll use proper form semantics in this example.
  - https://developer.mozilla.org/en/docs/Web/HTML/Element/form

ACTIVITY - Creating our Database Table

```
USE database_name;
CREATE TABLE artists (
  id int(11) NOT NULL AUTO_INCREMENT,
  name varchar(50) NOT NULL,
  founded_date date DEFAULT NULL,
  bio_link varchar(100) DEFAULT NULL,
  website varchar(100) DEFAULT NULL,
  PRIMARY KEY (id)
);
```

1. Connect to your DB with MySQL WorkBench
2. Enter the above SQL to create the new table

What is PDO?

What is it?

- PDO (PHP Data Objects)
- PDO provides a data access abstraction layer
    - supports several different database systems including PostgreSQL, SQLite, MS SQL Server, and MySQL
- PDO does not provide database abstraction (SQL rewriting)

Interactive Connection Exercise

http://comp1006-s16.azurewebsites.net/lesson-02/connection-exercise.html

ACTIVITY - Connecting to our Database

Example Files

/shared/secrets.php

/shared/connect.php

ACTIVITY - Using PHP to save to our Database

Example File:

/lesson-02/examples/add_artist.php