Assignment: Automobiles and Databases

In this assignment you will build a web based application to track data about automobiles and store the data in a MySQL database.

Sample solution

You can explore a sample solution for this problem at

Welcome to Autos Database

Please Log In

Attempt to go to autos.php without logging in - it should fail with an error message.

Specification for this Application

http://www.wa4e.com/solutions/autosdb/

Resources

There are several resources you might find useful:

- Recorded lectures, sample code and chapters from www.wa4e.com:
    - Review the SQL language
    - Using PDO in PHP

- Documentation on HTML Injection

- Documentation from www.php.net on how to use PDO to connect to a database.

- Documentation on SQL Injection

- Documentation on PHP PDO Prepared Statements

- You can look through the sample code from the lecture. It has examples of using PDO to communicate with a database:

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

    Note that this is not precisely sample code for *this* assignment. You should adapt your login code from the Rock Paper Scissors assignment using elements from the sample code above.

General Specifications

Here are some general specifications for this assignment:

- You **must** use the PHP PDO database layer for this assignment. If you use the "mysql_" library routines or "mysqli" routines to access the database, you will **receive a zero on this assignment**.

- Your name must be in the title tag of the HTML for all of the pages for this assignment.

- Your program must be resistant to HTML Injection attempts. All data that comes from the users must be properly escaped using the **htmlentities()** function in PHP. You do not need to escape text that is generated by your program.

- Your program must be resistant to SQL Injection attempts. This means that you should never concatenate user provided data with SQL to produce a query. You should always use a PDO prepared statement.

- Please do not use HTML5 in-browser data validation (i.e. type="number") for the fields in this assignment as we want to make sure you can properly do server side data validation. And in general, even when you do client-side data validation, you should still validate data on the server in case the user is using a non-HTML5 browser.

Databases and Tables Required for the Assignment

You already should have a PHP hosting environment such as MAMP or XAMPP installed or have some other access to a MySQL client to run commands.

You will need to create a database, a user to connect to the database and a password for that user using commands similar to the following:

```
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';
```

You will need to make a connection to that database in a file like this if you are using MAMP (Macintosh):

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

If you are using XAMPP or Linux you should change the port to 3306:

```php
<?php
$pdo = new PDO('mysql:host=localhost;port=3306;dbname=misc', 'fred', 'zap');
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

Usually this file is named `pdo.php` and is included in each of the files that want to use the database. You will need to change the user name and password on both your GRANT statements and in the code that makes the PDO connection.

You will also need to create and configure a table in the new "misc" database using the following SQL commands:

```sql
CREATE TABLE autos (
    auto_id INT UNSIGNED NOT NULL AUTO_INCREMENT KEY,
    make VARCHAR(128),
    year INTEGER,
    mileage INTEGER
);
```

Specifications

The changes to **index.php** are new wording and pointing to **autos.php** to test for login bypass.

Welcome to Autos Database

Please Log In

Attempt to go to autos.php without logging in - it should fail with an error message.

Specification for this Application

Specifications for the Login Screen

Much of the **login.php** is reused and extended from the previous assignment. The salt and hash computation and most of the error checking comes across unchanged. The password continues to be 'php123'.

The login screen needs to have some error checking on its input data. If either the name or the password field is blank, you should display a message of the form:

> Email and password are required

Note that we are using "email" and not "user name" to log in in this assignment.

If the password is non-blank and incorrect, you should put up a message of the form:

> Incorrect password

For this assignment, you must add one new validation to make sure that the login name contains an at-sign (@) and issue an error in that case:

> Email must have an at-sign (@)

If the incoming password, properly hashed matches the stored stored_hash value, the user's browser is [redirected](#) to the **autos.php** page with the user's name as a GET parameter using:

```
header("Location: autos.php?name=".urlencode($_POST['who']));
```

You must also use the **error_log()** function to issue the following message when the user fails login due to a bad password showing the computed hash of the password plus the salt:

```
error_log("Login fail ".$_POST['who']." $check");
```

When the login succeeds (i.e. the hash matches) issue the following log message:

```
error_log("Login success ".$_POST['who']);
```

Make sure to find your error log and find those error messages as they come out:

```
[11-Feb-2016 15:52:03 Europe/Berlin] Login success csev@autos.com
[11-Feb-2016 15:52:13 Europe/Berlin] Login fail csev@autos.com 047398bd0e0171f4954760f5f542121a
```

Specifications for the Auto Database Screen

In order to protect the database from being modified without the user properly logging in, the **autos.php** must first check the $_GET variable to see if the user's name is set and if the user's name is not present, the autos.php must stop immediately using the PHP die() function:

```
die("Name parameter missing");
```

To test, navigate to **autos.php** manually without logging in - it should fail with "Name parameter missing".

If the user is logged in, they should be presented with a screen that allows them to append a new make, mileage and year for an automobile. The list of all automobiles entered will be shown below the form. If there are no automobiles in the database, none need be shown.

If the **Logout** button is pressed the user should be redirected back to the **index.php** page using:

```
header('Location: index.php');
```

When the "Add" button is pressed, you need to do some input validation.

The mileage and year need to be integers. If is suggested that you use the PHP function **is_numeric()** to determine if the $_POST data is numeric. If either field is not nummeric, you must put up the following message:

Mileage and year must be numeric

Also if the make is empty (i.e. it has less than 1 character in the string) you need to put out a message as follows:

Make is required

Note that only one of the error messages need to come out regardless of how many errors the user makes in their input data. Once you detect one error in the input data, you can stop checking for further errors.
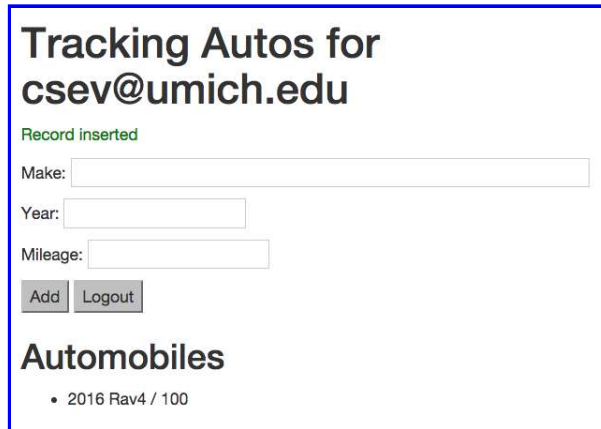
If the user has pressed the "Add" button and the data passes validation, you can add the automobile to the database using an **INSERT** statement.

```
    ...
    $stmt = $pdo->prepare('INSERT INTO autos
        (make, year, mileage) VALUES ( :mk, :yr, :mi)');
    $stmt->execute(array(
        ':mk' => $_POST['make'],
        ':yr' => $_POST['year'],
        ':mi' => $_POST['mileage'])
    );
    ...
```

When you successfully add data to your database, you need to put out a green "success message:

Record inserted

Once there are records in the database they should be shown below the form to add a new entry.

What To Hand In

For this assignment you will hand in:

1. A screen shot (including the URL) of your login.php rejecting an account without an at-sign (@)

2. A screen shot of your error log showing correct messages for both a successful and failed login attempt.

3. A screen shot (with URL) of your autos.php showing 'Name parameter missing'

4. A screen shot (including the URL) of your autos.php with three vehicles in the list. At least one of the vehicles must have '<b>' in its title and it must be shown properly (i.e. the title should not be bold)

5. A screen shot (including the URL) of your autos.php showing the error message for a non-numeric year

6. A screen shot of your autos database table in a database tool showing at least three vehicles

7. Source code of login.php

8. Source code of autos.php

Grading

Don't take off points for little mistakes. If they seem to have done the assignment give them full credit. Feel free to make suggestions if there are small mistakes. Please keep your comments positive and useful. If you do not take grading seriously, the instructors may delete your response and you will lose points.

The total number of points for this assignment is 10. You will get up to 5 points from your instructor. You will get up to 3 points from your peers. You will get 1 for each peer assignment you assess. You need to grade a minimum of 2 peer assignments. You can grade up to 5 peer assignments if you like.

Optional Challenges

This section is entirely *optional* and is here in case you want to explore a bit more deeply and test your code skillz.

Here are some possible improvements:

- Always show the automobiles sorted by make regardless of the order they were entered into the application. Hint: use "ORDER BY".

- Add an optional URL field to your tables and user interface. Validate the URL to make sure it starts with "http://" or "https://". If the user enters a URL, in the list of autos, have the make be a clickable anchor tag that opens the image in a new window:

  ```
  <a href="http://....jpg" target="_blank">Ford</a>
  ```

- Medium Difficulty: Use the PHP cURL library to do a GET to the image URL from within PHP and if the URL does not exist, issue an error message to the user and do not add the automobile.

Sample Database Screen Shots

The data in your screen shot(s) should not be the same as these examples.

Provided by: www.wa4e.com