

# WEEK 4 - Assignment Specification: Autos with Post-Redirect

---

In this next assignment, you will expand a web based application to track data about automobiles and store the data in a MySQL database. All interactions will follow the POST-Redirect pattern where appropriate. Note that there is no specific sample code for this assignment.

## Sample solution

You can explore a sample solution for this problem at <http://www.wa4e.com/solutions/autosess/>

## Resources

There are several resources you might find useful:

- Review the SQL language
- Using PDO in PHP
- Documentation on [Post-Redirect-GET Pattern](#)
- You can look though the sample code from the lecture. It has examples of using sessions and POST-Redirect: <https://www.wa4e.com/code/route.zip>

## 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

This assignment reuses the tables from the [previous assignment](#). No additional tables are necessary.

## Specifications

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

The basic functionality, password checking using salt and hashing, error logging, and data validation for the login.php is the same as in the [previous assignment](#).

There are several changes that are needed for this assignment as follows:

- The script must redirect after every POST. It must never produce HTML output as a result of a POST operation.

It must redirect to view.php instead of autos.php and must pass the logged in user's name through the session. A GET parameter is not allowed.

```
<?php // line added to turn on color syntax highlight

// Redirect the browser to view.php

$_SESSION['name'] = $_POST['email'];

header("Location: view.php");

return;
```

All error messages must be passed between the POST and GET using the session and "flash message" pattern:

```
<?php // line added to turn on color syntax highlight
```

```
$_SESSION['error'] = "Email must have an at-sign (@)";

header("Location: login.php");

return;
```

The error message must be displayed only on the next GET request.

```
<?php // line added to turn on color syntax highlight
if ( isset($_SESSION['error']) ) {
    echo('<p style="color: red;">'.htmlentities($_SESSION['error'])."</p>\n");
    unset($_SESSION['error']);
}
```

Subsequent GET requests (i.e. refreshing the page) should not show the error message to properly implement the POST-Redirect-GET-Flash pattern.

## Specifications for the Auto Database Screens

The **autos.php** script from the previous assignment is broken into two scripts in this assignment. The **view.php** script shows the list of automobiles in the database and the **add.php** script handles adding new automobiles to the database but does not list any autos. The **view.php** includes a link to **add.php** and **logout.php** and the **add.php** has a Cancel button.

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

```
<?php // line added to turn on color syntax highlight
if ( ! isset($_SESSION['name']) ) {
    die('Not logged in');
}
```

To test, navigate to **view.php** manually without logging in - it should fail with "Not logged in".

In **view.php** if the Logout button is pressed the user should be redirected back to the **logout.php** page. The **logout.php** page should clear the session and immediately redirect back to **index.php**:

```
<?php // line added to turn on color syntax highlight

session_start();
```

```
session_destroy();

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

In the **add.php** script, when the "Add" button is pressed, you need to do the same input validation as in the previous assignment, except that you must display the error using a proper POST-Redirect-GET-Flash pattern.

In the **add.php** script, when you successfully add data to your database, you need to redirect back to **view.php** and pass a "success message" to **view.php** using the session:

```
<?php // line added to turn on color syntax highlight

$_SESSION['success'] = "Record inserted";

header("Location: view.php");

return;
```

The **view.php** must detect and display the success message using the flash pattern:

```
<?php // line added to turn on color syntax highlight
if ( isset($_SESSION['success']) ) {
    echo('<p style="color: green;">'.htmlentities($_SESSION['success'])."</p>\n");
    unset($_SESSION['success']);
}
```

## What To Hand In

For this assignment you will hand in:

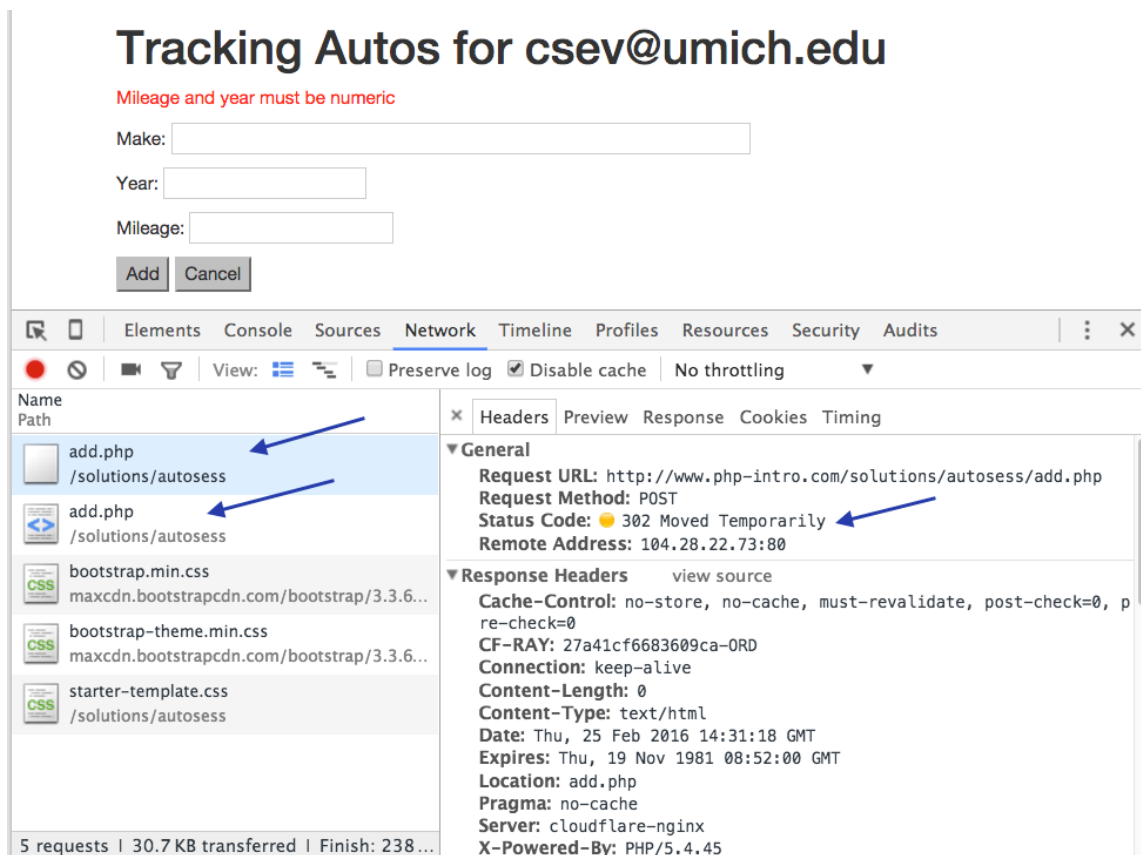
- A screen shot (including the URL) of your **login.php** rejecting an account without an at-sign (@). You must also include the developer console network tab showing both the POST and GET.
- A screen shot of your error log showing correct messages for both a successful and failed login attempt.
- A screen shot (with URL) of your **add.php** showing a data validation error. You must also include the developer console network tab showing both the POST and GET.
- A screen shot (including the URL) of your **view.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)

## 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. You need to grade a minimum of 2 peer assignments. You can grade up to 5 peer assignments if you like.

## Sample Screen Shots

Some of the screenshots ask to see the developer console demonstrating the POST-Redirect pattern similar to the following:



Some browsers don't actually show two separate requests for the post-redirect. Instead they show the POST and Redirect on the same line. They might produce a screen like the following:

# Please Log In

Email and password are required

User Name

Password

For a password hint, view source and find a password hint in the HTML comments.

The screenshot shows the Network tab of a web browser. The first request is a GET request to `login.php` with a status of 200 OK. A blue arrow points to this request with the label "GET (after redirect)". The second request is a POST request to `http://bf6ccf96.ngrok.io/...` with a status of "Redirect". A blue arrow points to this request with the label "POST / Redirect Cannot see headers :(". The bottom of the screenshot shows the console tab with a single log entry.

| Name      | Status | Type     | Initiator                    | Size   | Time   |
|-----------|--------|----------|------------------------------|--------|--------|
| login.php | 200 OK | document | http://bf6ccf96.ngrok.io/... | 1.1 KB | 178 ms |
| /hw8      |        |          | Redirect                     | 683 B  | 177 ms |

1 / 2 requests | 1.1 KB / 1.5 KB transferred | Finish: 178 ms | DOMContentLoaded: 254 ms | Load: 254 ms

Console

top ☐ Preserve log

Copyright Creative Commons Attribution 3.0 - Charles R. Severance