Name: Terrence Jackson
Date: 9.7.24
Course : CMSC 340 Web Programming
Week: 4
Assignment: Develop a PHP Script to Add and Display Course Records from a MYSQL Database Table

----

Rubric Criteria:
Number of correctly implemented specifications using PHP scripting out of five (5) specifications (50%)

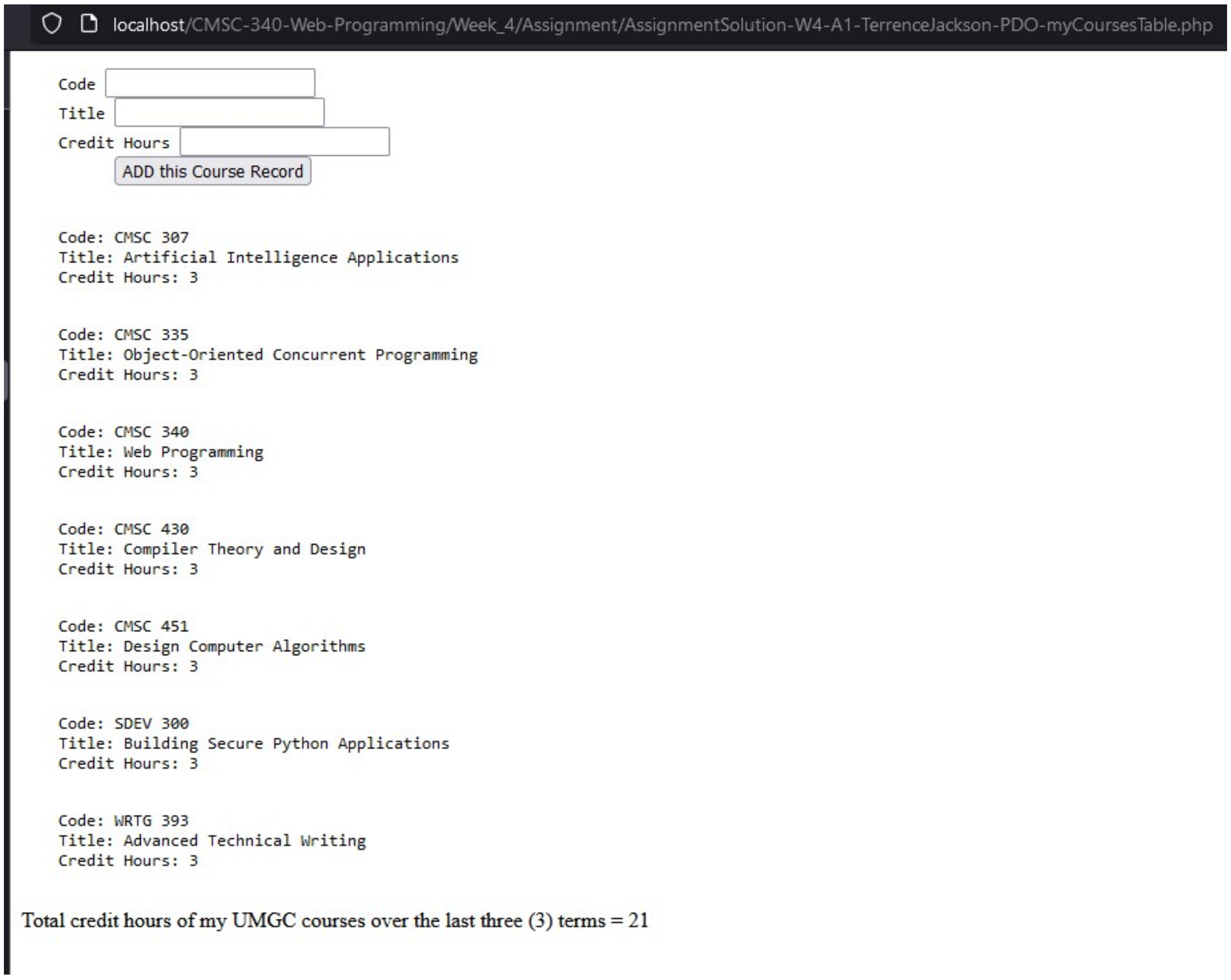
Embed here (or submit separately) a copy of your PHP script file(s):

Copy and paste here the text of your PHP code:

```
<?php

require_once 'login.php';


try {

  $pdo = new PDO($attr, $user, $pass, $opts);

} catch (PDOException $e) {

  throw new PDOException($e->getMessage(), (int) $e->getCode());

}


if (

  isset($_POST['credit_hours']) &&

  isset($_POST['title']) &&

  isset($_POST['code'])

) {

  $credit_hours = get_post($pdo, 'credit_hours');

  $title = get_post($pdo, 'title');

  $code = get_post($pdo, 'code');


  $query = "INSERT INTO mycourses VALUES" .

             "($code, $title, $credit_hours)";
  $result = $pdo->query($query);
}

echo <<<_END
  <form action="AssignmentSolution-W4-A1-TerrenceJackson-PDO-myCoursesTable.php"
method="post"><pre>
        Code <input type="text" name="code">
        Title <input type="text" name="title">
        Credit Hours <input type="text" name="credit_hours">
        <input type="submit" value="ADD this Course Record">
  </pre></form>
_END;

$query = "SELECT * FROM mycourses";
$result = $pdo->query($query);

while ($row = $result->fetch(PDO::FETCH_OBJ)) {
  $r0 = htmlspecialchars($row->code);
  $r1 = htmlspecialchars($row->title);
  $r2 = htmlspecialchars($row->credit_hours);

  echo <<<_END
  <pre>
        Code: $r0
        Title: $r1
        Credit Hours: $r2
  </pre>
_END;
}
```

```
$query = "SELECT SUM(credit_hours) as 'total' FROM mycourses";

$result = $pdo->query($query);

$result = $result->fetch();

$total = $result["total"];


echo "Total credit hours of my UMGC courses over the last three (3) terms = $total";


function get_post($pdo, $var)

{

  return $pdo->quote($_POST[$var]);

}
```

Insert here a screenshot(s) of the result of executing your PHP script:



localhost/CMSC-340-Web-Programming/Week_4/Assignment/AssignmentSolution-W4-A1-TerrenceJackson-PDO-myCoursesTable.php

```
Code
Title
Credit Hours
ADD this Course Record

Code: CMSC 307
Title: Artificial Intelligence Applications
Credit Hours: 3

Code: CMSC 335
Title: Object-Oriented Concurrent Programming
Credit Hours: 3

Code: CMSC 340
Title: Web Programming
Credit Hours: 3

Code: CMSC 430
Title: Compiler Theory and Design
Credit Hours: 3

Code: CMSC 451
Title: Design Computer Algorithms
Credit Hours: 3

Code: SDEV 300
Title: Building Secure Python Applications
Credit Hours: 3

Code: WRTG 393
Title: Advanced Technical Writing
Credit Hours: 3

Total credit hours of my UMGC courses over the last three (3) terms = 21
```

Rubric Criteria:
Number of relevant explanations out of three (3) explanations of approach, design, and steps of completing the assignment
(30%)

Your response here:

Approach: I began by using the SQL file provided in the week's discussion and adapted it for my specific database and table needs. My goal was to ensure the course_code could act as the primary key, so I avoided adding an auto-incrementing column. Once I adjusted the primary key handling, the database and table were created successfully.

Design: First, I used course_code as the primary key since it's unique, eliminating the need for an additional auto-incrementing field and keeping the table structure minimal. For user input, I used the basic HTML form that tied directly into the PHP script, focusing on simplicity without adding validation, since the requirements didn't call for it. For calculating total credit hours, I used SQL's SUM() function instead of handling the aggregation in PHP, making the process more efficient and leveraging the database's capabilities for better performance.

Steps:

1. **Database Creation**: I created a MySQL database called cmsc340 and a mycourses table. My initial challenge was setting course_code as the primary key, but once I removed the NULL default, everything worked fine.
2. **PHP Script:** Starting from previous discussion code, I modified the database connection and queries to fit the assignment's requirements. I also updated the form fields to accept course details and ensured proper handling of course addition with PDO. Debugging helped me refine the query that calculated the total credit hours using SUM().
3. **Testing the Data:** I populated the table with data from the last three semesters using the form, and confirmed that the data was inserted correctly.
4. **Final Formatting:** After ensuring functionality, I organized the code for clarity and ensured that the final output, including the total credit hours, appeared neatly for the screenshot.

Rubric Criteria:
Number of objective self-evaluations out of three (3) self-evaluations of strengths, weaknesses, and areas of improvement of self-performance on the assignment
(10%)

Your response here:

Strengths: One of my key strengths is problem-solving. When I encountered an issue with the primary key configuration for the course_code, I was able to quickly identify the problem and resolve it by adjusting the database schema. This ability to troubleshoot and resolve issues efficiently helped keep my project on track. Another strength is my adaptability. I successfully modified the provided SQL and PHP examples to suit the requirements of this assignment without much difficulty, demonstrating that I can adjust my approach when necessary. Lastly, I have a strong attention to detail. This was particularly evident when I ensured that the output met the exact format specified in the requirements, including taking extra steps to debug the total credit hour query until it displayed correctly.

Weaknesses: One of my weaknesses is that I initially struggled with the primary key setup. I mistakenly treated course_code like a regular column and spent unnecessary time troubleshooting before realizing it needed to be handled as a unique identifier. This oversight slowed down my progress. Another weakness is my tendency to over-focus on formatting. I can get caught up in making sure the code's aesthetics, such as spacing and indentation, are perfect, which sometimes takes time away from focusing on core functionality and overall efficiency.

Areas of Improvement: A key area where I can improve is in debugging efficiency. While I am capable of solving issues, I could benefit from refining my process for quickly identifying the root causes of problems. This would help me optimize my workflow and reduce the time spent troubleshooting. Additionally, I need to strike a better balance between formatting and core logic. While it's important to write clean, readable code, I should prioritize functionality and scalability first, then address formatting after ensuring everything works as intended.

Rubric Criteria:
Number of relevant reflections on the learning experience out of three (3) reflections of setbacks, triumphs, and lessons learned
10%

Your Response here:

Setbacks: I initially ran into an issue with the primary key configuration and later with the visibility of data in VSCode. These minor obstacles delayed my progress, but with some debugging, I overcame them.

Triumphs: The biggest win was implementing the total credit hour calculation using SQL's SUM() function. It took a bit of debugging, but once I got it working, the script behaved exactly as intended.

Lessons Learned: I learned to be more mindful when designing database schemas, especially regarding primary keys and their restrictions.