

Project Process Book

Isaac Loredo



Table of Contents

❖ Part 1	3
❖ Part 2	7
❖ Part 3	11
❖ Part 4	14
❖ Part 5	21
❖ Part 6	25

Part One

Process and Requirements

Welcome and Overview

Blueprint for the Web How it all works

Setting up Mamp and the PHP and MySQL environment

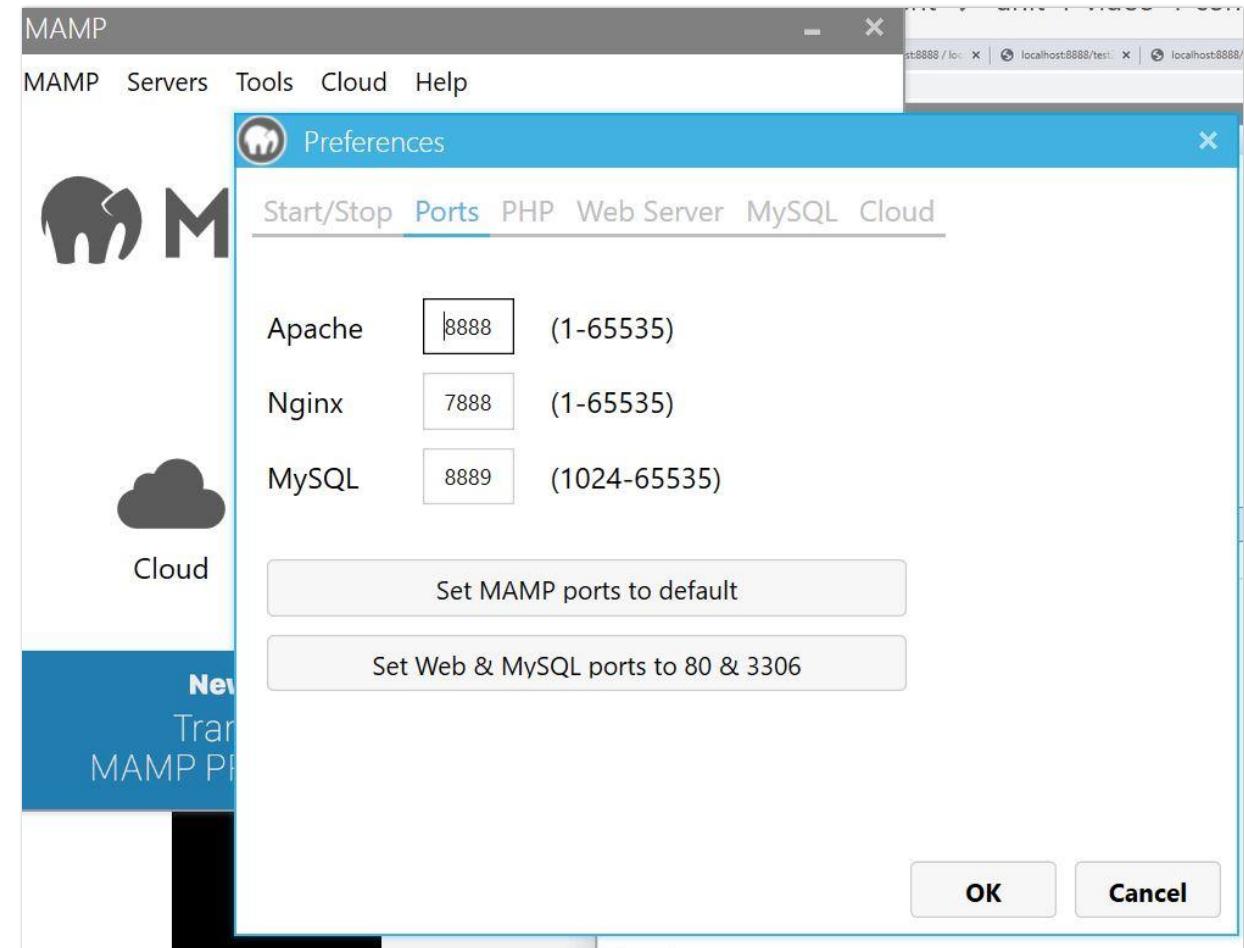
Using an IDE (Integrated Development Environment) for PHP development

Configuring MySQL and Setting up a database, database design and data types, database structure, database setup

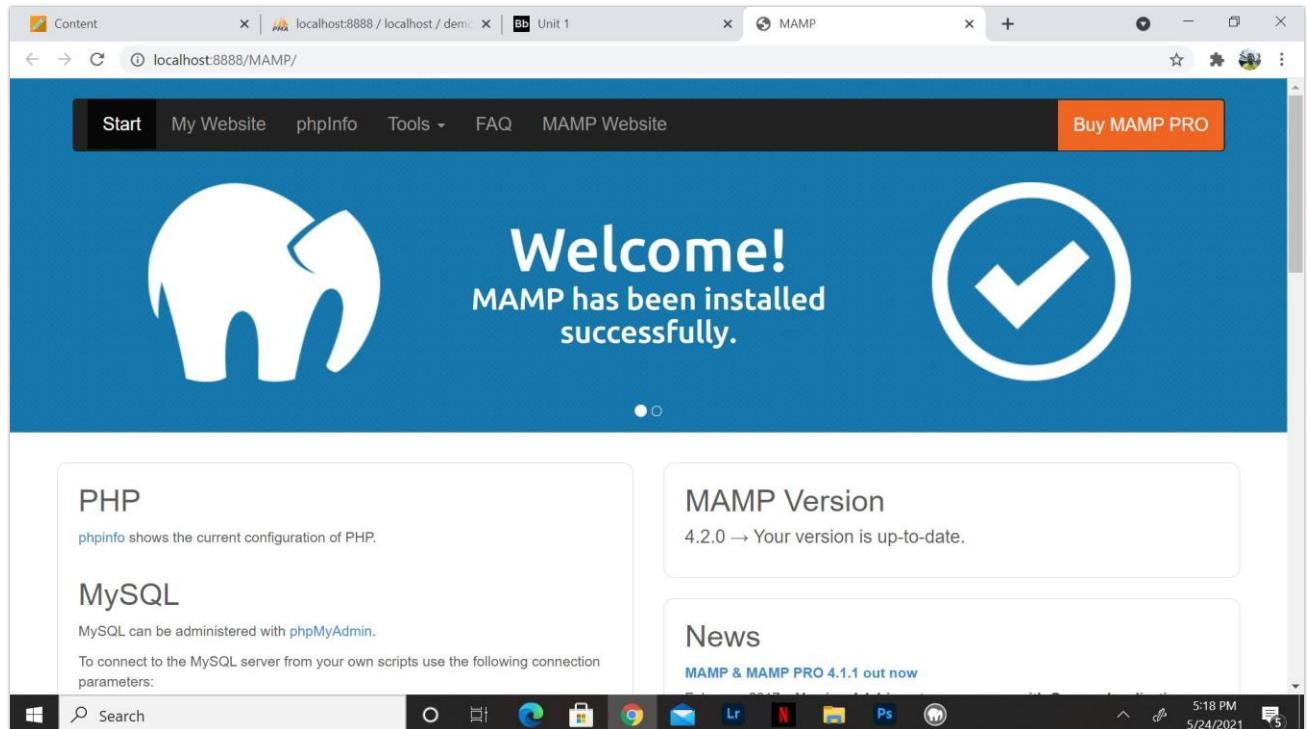
Connect PHP to MySQL

Connecting PHP to MySQL and Checking Ports

We began the project configuring the ports in MAMP.



After configuring the ports and settings in MAMP we verify the project has been linked successfully with the Open WebStart Page feature in MAMP which can also be accessed with the local host link that is connected to the Apache port.



Next, we created our database through MySQL, as well as our first table in that database and we altered the datatype in one of the columns as well. This was done through the phpMyAdmin tool which can be accessed through the localhost page.

The screenshot shows the phpMyAdmin interface for the 'content' table in the 'demo' database. The table has five columns: id, title, metaKeywords, metaDescription, and contentData. The 'id' column is defined as int(11) with AUTO_INCREMENT, while the other three are text type with utf8_general_ci collation. The 'contentData' column is marked as Primary. There is one index named 'PRIMARY' on the 'id' column. A note at the bottom states 'No partitioning defined!'. The interface includes tabs for Table structure, Relation view, and indexes/partitions.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	title	text	utf8_general_ci		No				Change Drop More
3	metaKeywords	text	utf8_general_ci		No				Change Drop More
4	metaDescription	text	utf8_general_ci		No				Change Drop More
5	contentData	text	utf8_general_ci		No				Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	2	A	No	

Create an index on columns

Partitions:

No partitioning defined!

Part Two

Process and Requirements

Continue to build on your PHP environment by creating your initial PHP files.

Initiate the process of creating a client side to server-side handoff Post methods.

Optionally integrate Twitter Bootstrap front-end framework for your CMS. This will not be shown throughout this class because this class is geared towards the back-end server-side logic. But you do have the option of making your CMS as a mobile-friendly CMS by integrating Twitter Bootstrap. This is shown in basic broad strokes here in this video and you will learn what to integrate and how to integrate it.

Create the HTML web form to capture the data.

Create a PHP file to catch the data from the form HTML form.

Create MySQL logic to select the database, authenticate the database, and insert the data into the database.

Check the data in the database via PHP MyAdmin.

Add security measure to prevent injection attacks.

Submit your additional narratives document containing your identified external learning and a description of your project enhancements.

Copy and Paste your Code into your Word Doc process book with clearly labeled sections so your professor can easily copy/paste the code out into their IDE.

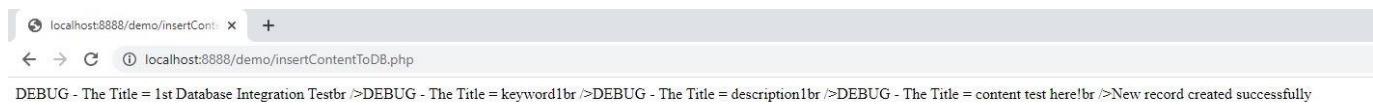
Screenshot all your code from your IDE with the file name clearly in view and insert the screenshots into your process book.

Zip all your programs and your process book and submit to the submission drop box.

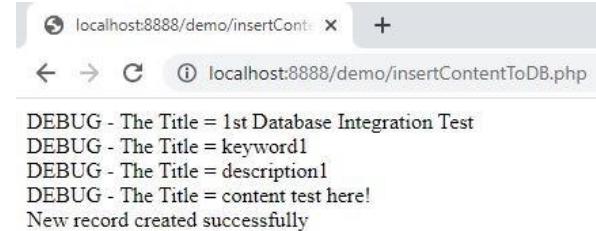
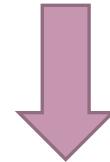
Submit your zip file into the submissions area for Project Part 2 by Day 7 of Week 2

When running the debug feature at the end I realized I made a mistake along the way it was not an error, however my results were not indented with the line break. Since it was not necessarily an error, I did not run it through the database. I proceeded with my problem-solving skills and after re looking at my code I realized I made an error and forgot the "<" before the break tags once I fixed this my problem was solved.

Problem I encountered



```
localhost:8888/demo/insertContentToDB.php
DEBUG - The Title = 1st Database Integration Test
DEBUG - The Title = keyword1
DEBUG - The Title = description1
DEBUG - The Title = content test here!
New record created successfully
```



```
localhost:8888/demo/insertContentToDB.php
DEBUG - The Title = 1st Database Integration Test
DEBUG - The Title = keyword1
DEBUG - The Title = description1
DEBUG - The Title = content test here!
New record created successfully
```

createContent.php screenshot

A screenshot of the Microsoft Visual Studio Code interface. The main editor area displays the following PHP code:

```
1 <form name="myForm" method="post" action="insertContentToDB.php"> <br />
2
3 <p>Enter a Title
4 <input type="text" name="title" id="title"> <br />
5 </p>
6 <p>Enter Keywords
7 <input type="text" name="metaKeywords" id="metaKeywords"> <br />
8 </p>
9 <p>Enter a Description
10 <input type="text" name="metaDescription" id="metaDescription"> <br />
11 </p>
12 <p>Enter your Content
13 <input type="text" name="contentData" id="contentData"> <br />
14 </p>
15
16
17 <input type="submit">
18 </form>
```

The code includes form fields for title, keywords, meta description, and content data, along with a submit button. The file is saved as `createContent.php`. The Solution Explorer on the right shows a folder named `Demo` containing `createContent.php` and `insertContentToDB.php`. The status bar at the bottom indicates the code has 1 issue found.

insertContentToDB.php screenshot

The screenshot shows a Microsoft Visual Studio Code interface with the following details:

- Title Bar:** File, Edit, View, Git, Tools, Extensions, Window, Help, Search (Ctrl+Q), Demo.
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Copy, along with navigation and search tools.
- Solution Explorer - Folder View:** Shows a folder named "Demo" containing "createContent.php" and "insertContentToDB.php".
- Code Editor:** The active tab is "insertContentToDB.php". The code is a PHP script for inserting content into a MySQL database. It includes comments explaining variable assignments and database connection parameters. It uses prepared statements to insert data into the "content" table.
- Status Bar:** Shows "50 %", "No issues found", "Ln: 52 Ch: 18 Col: 24 TABS CRLF", "Solution Explorer", "Git Changes", and a status bar icon.
- Taskbar:** Shows the Windows Start button, a search bar with "Type here to search", and pinned application icons for Edge, File Explorer, Mail, Lr, Netflix, Ps, and a purple icon.
- System Tray:** Shows the date and time as "7:25 PM 5/31/2021" and a notification icon with a red dot.

Part Three

1. Create the logic to select the database, authenticate the database, and select records from the database.
2. Reverse the security logic to strip out special characters that were inserted to prevent injection attacks.
3. Create a web page to render a list of records to the screen to select and combine this with the database selection logic.
4. Create a loop to iterate through all selected data and display that to the screen.
5. Create a query string parameter structure and append to the link of the record in order to pull up the specific record that was selected in the list of records.
6. Create a display page to display the information from the database to the screen after removing any data sanitation that was done for security measures.
7. Submit your additional narratives document containing your identified external learning and a description of your project enhancements.
8. Copy and Paste your Code into your Word Doc process book with clearly labeled sections so your professor can easily copy/paste the code out into their IDE.
9. Screenshot all your code from your IDE with the file name clearly in view and insert the screenshots into your process book.
10. Zip all your programs and your process book and submit to the submission drop box.
11. Submit your zip file into the submissions area for Project Part 3 by Day 7 of Week 3.

selectContentToModify.php code

The screenshot shows a Microsoft Visual Studio Code interface with the following details:

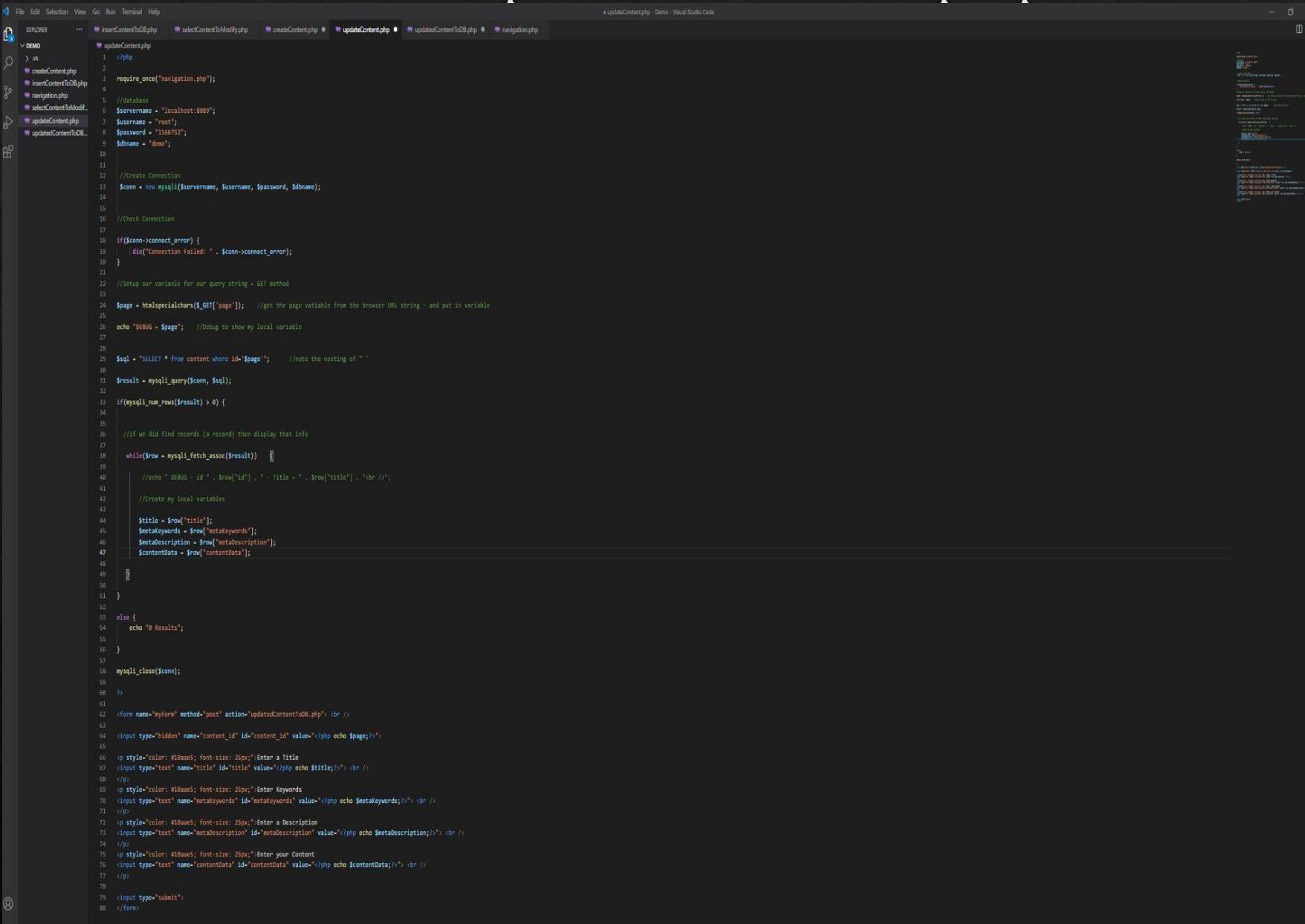
- Editor Area:** Displays the PHP code for `selectContentToModify.php`. The code connects to a MySQL database and outputs HTML table rows for each record found in the `content` table.
- Solution Explorer:** Shows a folder named `Demo` containing four files: `createContent.php`, `insertContentToDB.php`, and `selectContentToModify.php`.
- Taskbar:** At the bottom, it includes a search bar ("Type here to search"), a taskbar with icons for File Explorer, Task View, Edge, Google Chrome, Mail, Adobe Lightroom, Netflix, and Photoshop, and a system tray with a notification icon.
- Status Bar:** Shows "40 %", "No issues found", "Ln: 27 Ch: 1 MIXED CRLF", and the date/time "7:58 PM 6/7/2021".

```
1 <?php
2 //database
3 $servername = "localhost:8889";
4 $username = "root";
5 $password = "1160752";
6 $dbname = "demo";
7
8 //Create Connection
9 $conn = new mysqli($servername, $username, $password, $dbname);
10
11 //Check Connection
12 if($conn->connect_error) {
13     die("Connection Failed: " . $conn->connect_error);
14 }
15 //Method - Procedural
16 $sql = "SELECT * FROM content"; //select everything from the content table
17
18 if($result = mysqli_query($conn, $sql)) {
19
20     if(mysqli_num_rows($result) > 0) {
21
22         echo "<table>";
23         echo "<tr>";
24         echo "<th>id</th>";
25         echo "<th>title</th>";
26         echo "</tr>";
27
28         while($row = mysqli_fetch_array($result)) {
29             echo "<tr>";
30             echo "<td>" . "<a href=" . "updateContent.php?page=" . $row['id'] . ">" . $row['id'] . "</a>" . "</td>";
31             echo "<td>" . "<a href=" . "updateContent.php?page=" . $row['id'] . ">" . $row['title'] . "</a>" . "</td>";
32             echo "</tr>";
33
34         }
35
36         echo "</table>";
37         mysqli_free_result($result);
38     }
39
40     else {
41         echo "No records matching your query.";
42     }
43
44 }
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63     echo "ERROR: Could not execute $sql." . mysqli_error($conn);
64
65
66 }
67 mysqli_close($conn);
68 ?>
```

Part Four Update Records

- ❖ Integrate a global navigation menu system to tie your CMS programs together so they have to be loaded manually by their URL.
- ❖ Create a hidden form variable for database updates by primary key id
- ❖ Nest query strings with variables to execute SQL commands inside of PHP commands.
- ❖ Submit your additional narratives document containing your identified external learning and a description of your project enhancements.
- ❖ Copy and Paste your Code into your Word Doc process book with clearly labeled sections so your professor can easily copy/paste the code out into their IDE.
- ❖ Screenshot all your code from your IDE with the file name clearly in view and insert the screenshots into your process book.
- ❖ Zip all your programs and your process book and submit to the submission drop box.
- ❖ Submit your zip file into the submissions area for Project Part 4 by Day 7 of Week 4.

updateContent.php



The screenshot shows the Visual Studio Code interface with the file `updateContent.php` open in the main editor area. The code is a PHP script for updating content in a MySQL database. It includes connection logic, query execution, and a form for updating metadata and content.

```
<?php
require_once("navigation.php");
//database
$servername = "localhost:8889";
$username = "root";
$password = "1166752";
$dbname = "demo";

//Create Connection
$conn = new mysqli($servername, $username, $password, $dbname);

//Check Connection
if($conn->connect_error) {
    die("Connection Failed: " . $conn->connect_error);
}

//Setup our variable for our query string - GET method
$page = htmlspecialchars($_GET['page']); //get the page variable from the browser URL string - and put in variable
echo "DEBUG - $page"; //Debug to show my local variable

$sql = "SELECT * from content where id='$page';" //note the nesting of "
$result = mysqli_query($conn, $sql);

if(mysqli_num_rows($result) > 0) {

    //if we did find records (a record) then display that info
    while($row = mysqli_fetch_assoc($result)) {
        //echo "DEBUG - id " . $row["id"] . " - title " . $row["title"] . "  
";

        //Create my local variables
        $title = $row["title"];
        $metaKeywords = $row["metaKeywords"];
        $metaDescription = $row["metaDescription"];
        $contentData = $row["contentData"];

        ...
    }
} else {
    echo "0 Results";
}
mysqli_close($conn);
?>

<form name="myForm" method="post" action="updatedContentToDB.php">
<br/>
<input type="hidden" name="content_id" id="content_id" value="php echo $page; ?">
<br/>
<p style="color: #00aecc; font-size: 25px;">Enter a title
<input type="text" name="title" id="title" value="php echo $title; ?">
<br />
<p style="color: #00aecc; font-size: 25px;">Enter Keywords
<input type="text" name="metaKeywords" id="metaKeywords" value="php echo $metaKeywords; ?">
<br />
<p style="color: #00aecc; font-size: 25px;">Enter a Description
<input type="text" name="metaDescription" id="metaDescription" value="php echo $metaDescription; ?">
<br />
<p style="color: #00aecc; font-size: 25px;">Enter your Content
<input type="text" name="contentData" id="contentData" value="php echo $contentData; ?">
<br />
<br />
<input type="submit">
</form>
```

updateContent.php description

- ❖ We created a page to be able to update records into our database same way we did in the createContent.php file.
- ❖ However, this file was connected to the database we first utilized in the selectContentToModify.php from Part Three.
- ❖ The debugger displays the number of records in our data in our database which is the local variable.

updatedContent.php

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** updatedContentToDB.php - Demo - Visual Studio Code.
- Explorer:** Shows a folder named "DEMO" containing several PHP files: insertContentToDB.php, selectContentToModify.php, createContent.php, updateContent.php, updatedContentToDB.php, and navigation.php.
- Code Editor:** The main area displays the content of the "updatedContentToDB.php" file. The code is written in PHP and performs an UPDATE operation on a database table named "content". It handles POST variables for title, metaKeywords, metaDescription, and contentData, and performs HTML special character encoding on them. It then connects to a MySQL database using mysqli, checks the connection, and executes the update query. If successful, it outputs a message; if not, it outputs an error message.

```
1 <?php
2
3 require_once("navigation.php");
4
5 $id = htmlspecialchars($_POST['content_id']); //this is the primary key in the content db table to perform the UPDATE
6 $title = htmlspecialchars($_POST['title']);
7 $metaKeywords = htmlspecialchars($_POST['metaKeywords']);
8 $metaDescription = htmlspecialchars($_POST['metaDescription']);
9 $contentData = htmlspecialchars($_POST['contentData']);
10
11 //database
12 $servername = "localhost:8889";
13 $username = "root";
14 $password = "1166752";
15 $dbname = "demo";
16
17 //Create Connection
18 $conn = new mysqli($servername, $username, $password, $dbname);
19
20 //Check Connection
21
22 if($conn->connect_error) {
23     die("Connection failed: " . $conn->connect_error);
24 }
25
26 $sql = "UPDATE content SET title='$title', metaKeywords='$metaKeywords', metaDescription='$metaDescription', contentData='$contentData' WHERE id='$id'";
27
28 if(mysqli_query($conn, $sql)) {
29     echo "New record created successfully";
30 } else {
31     echo "Error: " . $sql . " " . mysqli_error($conn);
32 }
33
34 $conn->close();
35
36
37
38
39
40
41
42 ?>
```

updatedContent.php description

- ❖ This page is the landing page after successfully updating data into the database from the updateContent.php page.
- ❖ This file is also connected to the database and updates the input data.

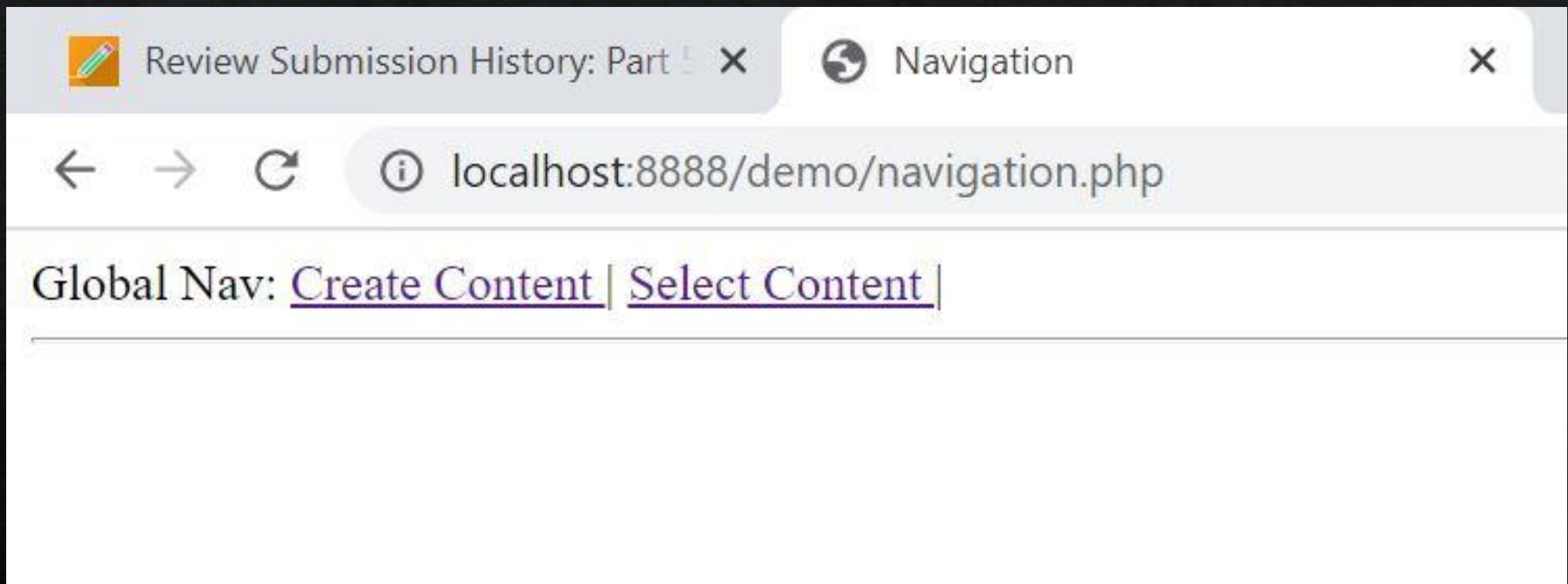
navigation.php

The screenshot shows the Visual Studio Code interface with the following details:

- Editor Area:** Displays the PHP file `navigation.php`. The code includes HTML and PHP logic for generating a navigation menu.
- Solution Explorer:** Shows a folder named `Demo` containing several PHP files: `createContent.php`, `deleteContent.php`, `insertContentToDB.php`, `navigation.php` (which is selected), `selectContentToModify.php`, `updateContent.php`, and `updatedContentToDB.php`.
- Status Bar:** Shows "100 %", "No issues found", "Ln: 1 Ch: 1 SPC CRLF", and a "Ready" status message.
- Bottom Right Corner:** A notification icon with a red circle containing the number "1".

```
navigation.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Navigation</title>
6 </head>
7 <body>
8 <?php
9
10 echo "Global Nav: ";
11 echo "<a href='createContent.php'>Create Content </a>" . " | ";
12 echo "<a href='selectContentToModify.php'>Select Content </a>" . " | ";
13
14 echo "<hr />";
15
16
17
18
19 <?>
20 </body>
21 </html>
```

navigation.php screenshot



navigation.php description

- ❖ We created a navigation menu to be able to access Create Content and Select Content without having to manually type in the URL in the browser.
- ❖ It is referencing all the other files in order to display the navigation menu.

Part 5

1. Learn how to leverage a hidden form variable for deleting specific information from the database.
2. Learn how to nest query strings with variables to execute a SQL command for deleting information from the database.
3. Learn how to integrate a global navigation to tie together pages in the CMS with a navigation system.
4. Learn how to display information from the database on a simple preview page.

deleteContent.php code

The screenshot shows a code editor with multiple tabs at the top: deleteContent.php (selected), navigation.php, selectContentToModify.php, createContent.php, and insertContentToDB.php. The deleteContent.php tab is active, displaying the following PHP code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Delete Content</title>
6 </head>
7 <body>
8 <?php
9
10 require_once("navigation.php");
11
12 //database
13 $servername = "localhost:8889";
14 $username = "root";
15 $password = "1166752";
16 $dbname = "demo";
17
18 //Create Connection
19 $conn = new mysqli($servername, $username, $password, $dbname);
20
21
22 //Check Connection
23
24 if($conn->connect_error) {
25   die("Connection Failed: " . $conn->connect_error);
26 }
27
28 $id = htmlspecialchars($_GET['page']);           //get the user id to delete from the database from the prior screen GET method
29
30 // sql to delete the record
31
32 $sql = "DELETE FROM content WHERE id='$id"';      //query string to use
33
34 if(mysqli_query($conn, $sql))  {
35   echo "Record deleted successfully.";
36 }
37
38 else {
39   echo "Error deleting record: " . mysqli_error($conn);
40 }
41
42
43 mysqli_close($conn);
44
45 ?>
46 </body>
47 </html>
```

deleteContent.php screenshot

Assignment Submissions – 2021S | localhost:8888/demo/selectContentToModify.php | PHP Code Checker - Syntax Checker

Global Nav: [Create Content](#) | [Select Content](#)

id	Title	Delete Content
1	Test	Delete Content
2	Test 2	Delete Content
3	1st Database Integration Test	Delete Content
4	1st Database Integration Test	Delete Content
5	1st Database Integration Test	Delete Content
7	8th Database Integration Test	Delete Content
8		Delete Content

Assignment Submissions – 2021S | localhost:8888/demo/deleteContent.php | PHP Code Checker - Syntax Checker

Global Nav: [Create Content](#) | [Select Content](#)

Record deleted successfully.

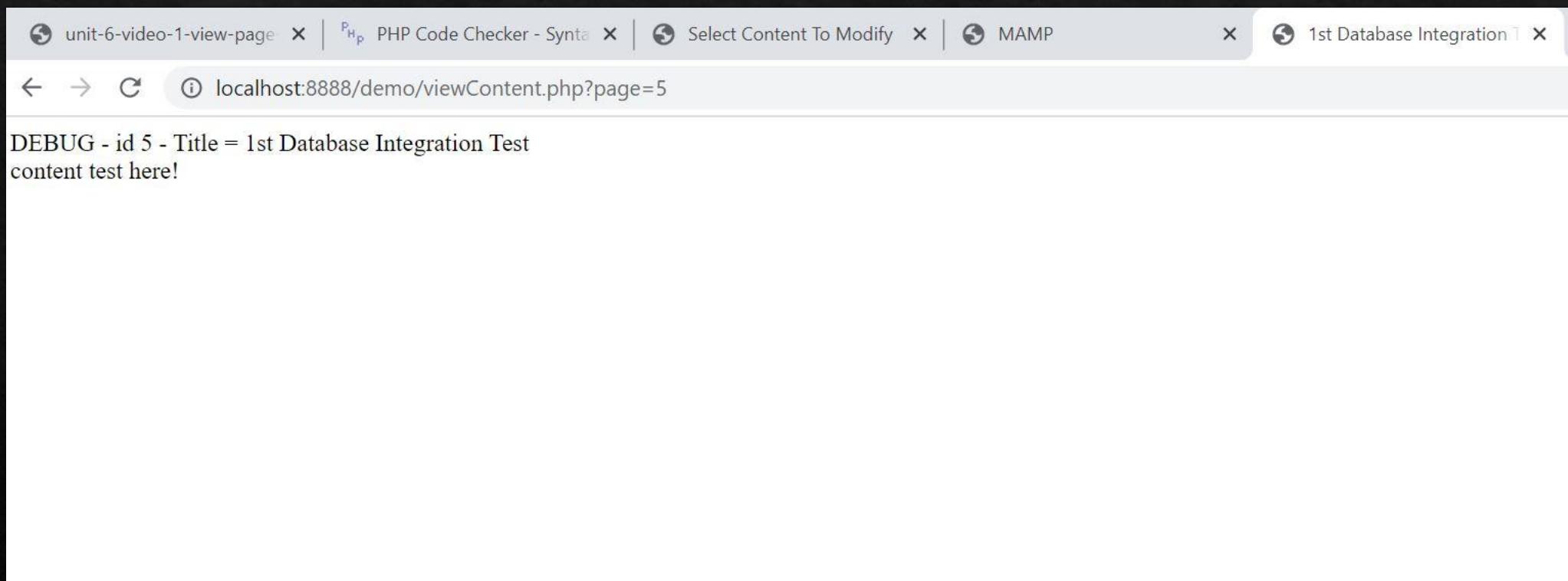
deleteContent.php description

- ❖ We added a Delete Content column and heading in the selectContentToModify.php file.
- ❖ In addition, we created a new file with previously utilized code where we obtained the user ID in order to delete data from the prior GET method screen.
- ❖ Utilizing an SQL command, we set it to delete a single column at a time with a query string as well obtained from the ID.
- ❖ Finally, we added a message to display to determine whether if the deletion was successful and this page is linked to the navigation as well.

Part 6

- ❖ Create the programming logic to select a specific record from the database and store it into a PHP array.
- ❖ Render out a dynamic HTML page that displays data from the database via PHP.
- ❖ Copy and Paste your Code into your Word Doc process book with clearly labeled sections so your professor can easily copy/paste the code out into their IDE.
- ❖ Submit your additional narratives document containing your identified external learning and a description of your project enhancements.
- ❖ Screenshot all your code from your IDE with the file name clearly in view and insert the screenshots into your process book.
- ❖ Create a one to three minute video demo of your project that demonstrates how each of the operations work and then demo your final display page's final output and HTML output as shown in the lecture video.
- ❖ Zip all your programs and your process book, and your video and submit to the submission drop box.
- ❖ Submit your zip file into the submissions area for Project Part 6 by Day 3 of Week 6.

viewContent.php screenshot



viewContent.php description

- ❖ We completed the CRUD model creating a display page in our content management system.
- ❖ This allows the user to display specific records from the database.