**E-COMMERCE**

420-411-VA

#### **ASSIGNMENT-1-PART-1**

Since we need to use PHP as a back-end implementation language, we need to learn about the PHP language.

For the parts that require code implementation, write the code in specific files and submit the files as a compressed file with this document.

1. Describe the need for and the usage of the PHP language:

|  |
| --- |
| Create dynamic and interactive web applications by executing server-side scripts, handling data processing, and facilitating seamless integration with databases and web frameworks. |

1. Is PHP a strongly typed language? Do we need to assign a type to variable before its usage?:

|  |
| --- |
| No. it is dynamically typed. No, you do not need to assign a type to a variable before using it. |

1. What are the data types in PHP?

|  |
| --- |
| String, Integer, Float “also called double, Boolean, Array, Object, NULL, Resource |

1. How do we define a string in PHP? and what is the difference between a double quoted string "value" and a single quoted string, give an example?

|  |
| --- |
| you can define a string using either double quotes (") or single quotes (').  Double-quoted strings allow variable interpolation and interpret escape sequences, while single-quoted strings treat most characters as literal and do not interpolate variables.  For example:  $name = "Dolphin";  echo "Hello, $name!"; // Output: Hello, Dolphin!  $name = "Unicorn";  echo 'Hello, $name!'; // Output: Hello, $name! |

1. What are the different types of operators in PHP?

|  |
| --- |
| Arithmetic operators, Assignment operators,  Comparison operators, Increment/Decrement operators,  Logical operators, String operators,  Array operators, Conditional assignment operators |

1. What is the difference between the “==” and “===” operator, give an example?

|  |
| --- |
| the "==" operator checks for equality of values, possibly performing type coercion, and  The "===" operator checks for strict equality, ensuring both values and data types of matches.  For Example:  5 == "5" // is true  5 === "5" // is false |

1. What are the naming conventions in PHP for variables, functions, and classes?

|  |
| --- |
| Variables: start with $ dollar sign and use lowercase letters and underscores to separate words.  Functions: use lowercase letters and underscores to separate words.  Classes: class name starts with an uppercase letter, and there are no underscores or spaces. |

1. What is the difference between “echo”, “print”, and “var\_dump”?

|  |
| --- |
| 1. echo:    * Used for displaying content to the screen, such as HTML and text.    * Can output multiple expressions separated by commas.    * Has no return value, so it cannot be used in assignments.    * Commonly used for quick and straightforward output. 2. print:    * Also used for displaying content to the screen.    * Can only output a single expression.    * Returns a value of 1, allowing it to be used in assignments, although it's less commonly used for this purpose.    * Typically used when you need to display a single value and capture its return value simultaneously. 3. var\_dump:    * Used for debugging and inspecting variables.    * Displays detailed information about variable data types and values, making it helpful during development and troubleshooting.    * *Not used for typical content output; instead,* it provides insights into the structure and contents of variables and arrays. |

1. What is the use of the “isset” function, give an example?

|  |
| --- |
| The “isset” function is used to determine whether a variable is set and is not NULL.  $name = "Dolphin";  isset($name) // will return true!  $name1;  isset($name1) // will return false! |

1. What is the use of the “isempty” function, give an example?

|  |
| --- |
| **isempty** is not a built-in *function* in PHP. However, isempty can be method for other classes in php such as DsPair. |

1. What is the use of the “include” statement? Write an example showing how the content of one PHP file echo data when included in another file:

|  |
| --- |
| It is used to include the content of one PHP file into another PHP file. When a file is included, the code it contains is evaluated and executed as if it were part of the calling file.  // File: name.php  <?php  $name = "Dolphin";  ?>  // File: index.php  <?php  include name.php';  echo $ name;  ?>  // Dolphin! |

1. What is the use of the “require” statement? and “require\_once”?

|  |
| --- |
| The **require** statement is used to include the content of one PHP file into another PHP file.  The **require\_once** statement is like the require statement, but with an additional check: if the specified file has already been included, it will not be included again. |

1. What is the difference between the “require” and “include” statements?

|  |
| --- |
| The main difference is how they handle **errors** when the specified file is not found.  if the file specified in the **include** statement is not found, a warning is generated, but the script will continue to execute.  f the file specified in the **require** statement is not found, a fatal error is generated, and the script execution is halted. |

1. How do we echo the path of the file containing the current script?

|  |
| --- |
| <?php  echo \_\_DIR\_\_  ?> |

1. How do we echo the path of the parent directory of the file containing the current script?

|  |
| --- |
| <?php  echo dirname(\_\_DIR\_\_);  ?> |

1. What are superglobal variables in PHP? Write an example using the “$\_SERVER” superglobal array:

|  |
| --- |
| Superglobal variables in PHP are built-in variables that are always available in all scopes.  Superglobal variables are:   * $GLOBALS * $\_SERVER * $\_GET * $\_POST * $\_FILES * $\_COOKIE * $\_SESSION * $\_REQUEST * $\_ENV   <?php  echo $\_SERVER['PHP\_SELF'];  echo "<br>";  echo $\_SERVER['SERVER\_NAME'];  echo "<br>";  echo $\_SERVER['HTTP\_HOST'];  echo "<br>";  echo $\_SERVER['HTTP\_REFERER'];  echo "<br>";  echo $\_SERVER['HTTP\_USER\_AGENT'];  echo "<br>";  echo $\_SERVER['SCRIPT\_NAME'];  ?> |

1. How would you define a constant in PHP:

|  |
| --- |
| you can define a constant using the **define()** function or the **const** keyword.  <?php    const MY\_CONSTANT1 = 'Hello, World!';  define('MY\_CONSTANT2', 'Hello, PHP!');    echo MY\_CONSTANT1;  echo MY\_CONSTANT2;  ?> |

1. How to define a global variable in PHP accessible from multiple scopes?

|  |
| --- |
| By using the **global** keyword or the $GLOBALS superglobal array.  <?php  $my\_global\_var1 = 'Hello, World!';  $my\_global\_var2 = 'Hello, PHP!';  function my\_function() {  global $my\_global\_var1;  echo $my\_global\_var1;  echo $GLOBALS['my\_global\_var2'];  }  my\_function();  ?> |

1. How to read values from an HTML form using PHP, give an example?

|  |
| --- |
| using PHP by accessing the $\_POST or $\_GET superglobal arrays.  <!-- File: form.html -->  <form action="submit.php" method="post">  <label for="name">Name:</label>  <input type="text" id="name" name="name"><br>  <label for="email">Email:</label>  <input type="email" id="email" name="email"><br>  <input type="submit" value="Submit">  </form>  <?php  //the keys corresponding to the name attributes of the form fields  $name = $\_POST['name'];  $email = $\_POST['email'];  echo "Name: " . $name . "<br>";  echo "Email: " . $email;  ?> |

1. How to read data from a database and display it on a webpage using MariaDB and PHP?
   1. Create a database
   2. Create a table called employees and fill-in some sample data
   3. Create the PHP code that connects to the database using the PDO library
   4. Select the data from the database using a SELECT query
   5. Write a foreach loop that loops over the data and build an HTML table
   6. Display the HTML table on a webpage

|  |
| --- |
| **b.**  **// companydb.sql**  CREATE TABLE employees (  id INT AUTO\_INCREMENT PRIMARY KEY,  first\_name VARCHAR(50),  last\_name VARCHAR(50),  email VARCHAR(100)  );  INSERT INTO employees (first\_name, last\_name, email)  VALUES  ('John', 'Doe', 'john.doe@example.com'),  ('Jane', 'Smith', 'jane.smith@example.com'),  ('Bob', 'Johnson', 'bob.johnson@example.com');  **c.**  **//** **database.php**  **<?php**  $hostname = "localhost";  $username = "admin";  $password = "";  $database = "companydb";  try {  $db = new PDO("mysql:host=$hostname;dbname=$database", $username, $password);  $db->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);  echo "Connected successfully";  } catch (PDOException $e) {  echo "Connection failed: " . $e->getMessage();  }  **d.**  $query = "SELECT \* FROM employees";  $stmt = $db->query($query);  $employees = $stmt->fetchAll(PDO::FETCH\_ASSOC);  ?>  **e.**  <table>  <tr>  <th>ID</th>  <th>First Name</th>  <th>Last Name</th>  <th>Email</th>  </tr>  <?php foreach ($employees as $employee) : ?>  <tr>  <td><?php echo $employee['id']; ?></td>  <td><?php echo $employee['first\_name']; ?></td>  <td><?php echo $employee['last\_name']; ?></td>  <td><?php echo $employee['email']; ?></td>  </tr>  <?php endforeach; ?>  </table>  **f.**  <!DOCTYPE html>  <html>  <head>  <title>Employee List</title>  </head>  <body>  <h1>Employee List</h1>  <?php include("database.php"); ?>  </body>  </html> |