**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Topic** | **Date** | **Teacher's Signature** |
| 1 | Question 1 |  |  |
| 2 | Question 2 |  |  |
| 3 | Question 3 |  |  |
| 4 | Question 4 |  |  |
| 5 | Question 5 |  |  |
| 6 | Question 6 |  |  |
| 7 | Question 7 |  |  |
| 8 | Question 8 |  |  |
| 9 | Question 9 |  |  |

**Question 1**

Source Code

|  |
| --- |
| <?php /\* Q1. Write a regular expression including modifiers, operators and Metacharacters (Menu driven) \*/  $result = "";  if(isset($\_POST['submit'])){  $choice = $\_POST['choice'];  $input = $\_POST['input'];   switch($choice){  case 1: // ^ and $ (start and end)  if(preg\_match("/^hello$/", $input)){  $result = "Matched exactly 'hello'";  } else {  $result = "No match";  }  break;   case 2: // . (any single character)  if(preg\_match("/h.llo/", $input)){  $result = "Matched pattern h.llo (any single char in place of .)";  } else {  $result = "No match";  }  break;   case 3: // \* and + (zero or more, one or more)  if(preg\_match("/ab\*c/", $input)){  $result = "Matched pattern ab\*c (zero or more b)";  } elseif(preg\_match("/ab+c/", $input)){  $result = "Matched pattern ab+c (one or more b)";  } else {  $result = "No match";  }  break;   case 4: // | OR and () grouping  if(preg\_match("/cat|dog/", $input)){  $result = "Matched cat or dog";  } elseif(preg\_match("/(red|blue)car/", $input)){  $result = "Matched redcar or bluecar";  } else {  $result = "No match";  }  break;   case 5: // Modifiers i, m, s  if(preg\_match("/hello/i", $input)){  $result = "Matched 'hello' case-insensitive (i)";  } elseif(preg\_match("/^start/m", $input)){  $result = "Matched 'start' at beginning of line (m)";  } elseif(preg\_match("/first.\*last/s", $input)){  $result = "Matched across lines using dotall (s)";  } else {  $result = "No match with modifiers";  }  break;   default:  $result = "Please select a valid option.";  } } ?>  <!DOCTYPE html> <html> <head>  <title>Regex Menu Driven Program</title> </head> <body>  <h2>Regex Menu Driven Program</h2>   <form method="post">  <label>Select Option:</label>  <select name="choice">  <option value="1">^ and $ (Start/End of string)</option>  <option value="2">. (Any single character)</option>  <option value="3">\* and + (Zero or more, One or more)</option>  <option value="4">| (OR) and () grouping</option>  <option value="5">Modifiers: i, m, s</option>  </select><br><br>   <label>Enter Input:</label>  <input type="text" name="input" required><br><br>   <input type="submit" name="submit" value="Check">  </form>   <?php  if($result != ""){  echo "<h3>Result: $result</h3>";  }  ?> </body> </html> |

Output

|  |
| --- |
| {'usageMetadata': {'promptTokenCount': 1136, 'totalTokenCount': 1136, 'promptTokensDetails': [{'modality': 'TEXT', 'tokenCount': 1136}]}, 'modelVersion': 'gemini-2.0-flash-exp', 'responseId': '0jjZaMDfE67H0-kPvdC-8A8'} |

**Question 2**

Source Code

|  |
| --- |
| <?php /\* Q2. Write a program to show the usage of switch-case and for/while/do while loop to create a menu driven program for calculator \*/  $result = "";  if(isset($\_POST['submit'])){  $num1 = $\_POST['num1'];  $num2 = $\_POST['num2'];  $choice = $\_POST['choice'];   switch($choice){  case "1":  $result = $num1 + $num2;  break;  case "2":  $result = $num1 - $num2;  break;  case "3":  $result = $num1 \* $num2;  break;  case "4":  if($num2 != 0){  $result = $num1 / $num2;  } else {  $result = "Error: Division by zero!";  }  break;  default:  $result = "Invalid choice!";  } } ?>  <!DOCTYPE html> <html> <head>  <title>Simple Calculator</title> </head> <body>  <h2>Simple Calculator</h2>  <form method="post">  <label>First Number:</label>  <input type="number" name="num1" required><br><br>   <label>Second Number:</label>  <input type="number" name="num2" required><br><br>   <label>Choose Operation:</label>  <select name="choice">  <option value="1">Addition</option>  <option value="2">Subtraction</option>  <option value="3">Multiplication</option>  <option value="4">Division</option>  </select><br><br>   <input type="submit" name="submit" value="Calculate">  </form>   <?php  if($result !== ""){  echo "<h3>Result: $result</h3>";  }  ?> </body> </html> |

Output

|  |
| --- |
| {'usageMetadata': {'promptTokenCount': 802, 'totalTokenCount': 802, 'promptTokensDetails': [{'modality': 'TEXT', 'tokenCount': 802}]}, 'modelVersion': 'gemini-2.0-flash-exp', 'responseId': '1TjZaNjYEriBmtkPwuDeoQk'} |

**Question 3**

Source Code

|  |
| --- |
| <?php /\* Q3. Menu-driven program to perform all types of array sorting \*/  if(isset($\_POST['submit'])){  // Input array (comma-separated)  $input = $\_POST['numbers'];  $numbers = array\_map('trim', explode(",", $input)); // Convert to array   // Associative array (predefined for demonstration)  $assoc = array("a"=>3, "b"=>1, "c"=>4, "d"=>2);   $continue = true;  $output = "";   while($continue){  $choice = $\_POST['choice'] ?? "";   switch($choice){  case "1": // Ascending sort  $asc = $numbers;  sort($asc);  $output = "Ascending Sort: ";  $output .= implode(", ", $asc);  break;   case "2": // Descending sort  $desc = $numbers;  rsort($desc);  $output = "Descending Sort: ";  $output .= implode(", ", $desc);  break;   case "3": // Associative sort by values ascending  $assoc\_copy = $assoc;  asort($assoc\_copy);  $output = "Associative Array Ascending by values: ";  foreach($assoc\_copy as $k => $v) $output .= "$k=>$v, ";  $output = rtrim($output, ", ");  break;   case "4": // Associative sort by values descending  $assoc\_copy = $assoc;  arsort($assoc\_copy);  $output = "Associative Array Descending by values: ";  foreach($assoc\_copy as $k => $v) $output .= "$k=>$v, ";  $output = rtrim($output, ", ");  break;   case "5": // Sort by keys ascending  $assoc\_copy = $assoc;  ksort($assoc\_copy);  $output = "Associative Array Ascending by keys: ";  foreach($assoc\_copy as $k => $v) $output .= "$k=>$v, ";  $output = rtrim($output, ", ");  break;   case "6": // Sort by keys descending  $assoc\_copy = $assoc;  krsort($assoc\_copy);  $output = "Associative Array Descending by keys: ";  foreach($assoc\_copy as $k => $v) $output .= "$k=>$v, ";  $output = rtrim($output, ", ");  break;   case "7": // Exit  $continue = false;  $output = "Exited menu.";  break;   default:  $output = "Please select a valid option.";  }   break; // Break the while loop after processing choice once  } } ?>  <!DOCTYPE html> <html> <head>  <title>Array Sorting Demo - Menu Driven</title> </head> <body>  <h2>Array Sorting Demo (Menu Driven)</h2>   <form method="post">  <label>Enter numbers (comma-separated):</label>  <input type="text" name="numbers" required value="<?php echo $\_POST['numbers'] ?? ''; ?>"><br><br>   <label>Select sorting operation:</label><br>  <input type="radio" name="choice" value="1"> Sort Numbers Ascending<br>  <input type="radio" name="choice" value="2"> Sort Numbers Descending<br>  <input type="radio" name="choice" value="3"> Associative Ascending by values<br>  <input type="radio" name="choice" value="4"> Associative Descending by values<br>  <input type="radio" name="choice" value="5"> Associative Ascending by keys<br>  <input type="radio" name="choice" value="6"> Associative Descending by keys<br>  <input type="radio" name="choice" value="7"> Exit<br><br>   <input type="submit" name="submit" value="Submit">  </form>   <?php if(isset($numbers)) { ?>  <h3>Original Array:</h3>  <?php echo implode(", ", $numbers); ?>   <h3>Associative Array:</h3>  <?php foreach($assoc as $k=>$v) echo "$k=>$v, "; ?>   <h3>Result:</h3>  <?php echo $output; ?>  <?php } ?> </body> </html> |

Output

|  |
| --- |
| {'usageMetadata': {'promptTokenCount': 1375, 'totalTokenCount': 1375, 'promptTokensDetails': [{'modality': 'TEXT', 'tokenCount': 1375}]}, 'modelVersion': 'gemini-2.0-flash-exp', 'responseId': '2DjZaICNK6KwqtsPvoKRiQk'} |

**Question 4**

Source Code

|  |
| --- |
| <?php /\* Q4. Menu-driven program to implement Array-pad(), array\_slice(), array\_splice(), list() functions. \*/  $result = "";  if(isset($\_POST['submit'])){  // Get input array from form (comma-separated)  $input = $\_POST['array'];  $arr = explode(",", $input); // convert to array  $continue = true;  $output = ""; // store results for display   while($continue){  $choice = $\_POST['choice'] ?? "";   switch($choice){  case "1": // array\_pad  $padded = array\_pad($arr, 7, "X");  $output = "Array after array\_pad (length 7, pad with 'X'):<br>";  foreach($padded as $value) $output .= $value . " ";  break;   case "2": // array\_slice  $sliced = array\_slice($arr, 1, 3);  $output = "Array after array\_slice (3 elements from index 1):<br>";  foreach($sliced as $value) $output .= $value . " ";  break;   case "3": // array\_splice  $splice\_arr = $arr; // copy array  array\_splice($splice\_arr, 2, 2, ["Y","Z"]);  $output = "Array after array\_splice (replace 2 elements at index 2 with 'Y','Z'):<br>";  foreach($splice\_arr as $value) $output .= $value . " ";  break;   case "4": // list()  list($first, $second) = $arr;  $output = "Using list() to get first two elements:<br>";  $output .= "First: $first, Second: $second";  break;   case "5": // Exit  $continue = false;  $output = "Exited menu.";  break;   default:  $output = "Please select a valid option.";  }   break; // break the while after processing choice once  } } ?>  <!DOCTYPE html> <html> <head>  <title>Array Functions Demo - Menu Driven</title> </head> <body>  <h2>Array Functions Demo (Menu Driven)</h2>   <form method="post">  <label>Enter array elements (comma-separated):</label>  <input type="text" name="array" required value="<?php echo $\_POST['array'] ?? ''; ?>"><br><br>   <label>Select operation:</label><br>  <input type="radio" name="choice" value="1"> Array Pad<br>  <input type="radio" name="choice" value="2"> Array Slice<br>  <input type="radio" name="choice" value="3"> Array Splice<br>  <input type="radio" name="choice" value="4"> List<br>  <input type="radio" name="choice" value="5"> Exit<br><br>   <input type="submit" name="submit" value="Submit">  </form>   <?php if(isset($arr)) { ?>  <h3>Original Array:</h3>  <?php foreach($arr as $value) echo $value . " "; ?>   <h3>Result:</h3>  <?php echo $output; ?>  <?php } ?> </body> </html> |

Output

|  |
| --- |
| {'usageMetadata': {'promptTokenCount': 1126, 'totalTokenCount': 1126, 'promptTokensDetails': [{'modality': 'TEXT', 'tokenCount': 1126}]}, 'modelVersion': 'gemini-2.0-flash-exp', 'responseId': '2zjZaMyALO-ez7IP2-\_nqQ8'} |

**Question 5**

Source Code

|  |
| --- |
| <?php /\* Q5. Menu-driven Login Form Validation using PHP \*/  $message = "";  // Predefined credentials for demo $valid\_username = "user123"; $valid\_password = "Pass@123";  if(isset($\_POST['submit'])){  $continue = true;  $choice = $\_POST['choice'] ?? "";   while($continue){  switch($choice){  case "1": // Login validation  // Using $\_REQUEST to get data  $username = $\_REQUEST['username'] ?? "";  $password = $\_REQUEST['password'] ?? "";   // Validate username: only letters and numbers, 3-15 characters  if(!preg\_match("/^[a-zA-Z0-9]{3,15}$/", $username)){  $message = "Invalid username! Only letters/numbers, 3-15 chars.";  }  // Validate password: at least 6 chars, must include letters & numbers  elseif(!preg\_match("/^(?=.\*[A-Za-z])(?=.\*\d)[A-Za-z\d@#$%^&+=]{6,}$/", $password)){  $message = "Invalid password! Must be at least 6 chars with letters & numbers.";  }  // Check credentials  elseif($username == $valid\_username && $password == $valid\_password){  $message = "Login successful! Welcome $username.";  } else {  $message = "Login failed! Incorrect username or password.";  }  break;   case "2": // Display credential rules  $message = "Username: 3-15 chars, letters & numbers only.<br>";  $message .= "Password: At least 6 chars, must include letters & numbers.";  break;   case "3": // Exit  $continue = false;  $message = "Exited menu.";  break;   default:  $message = "Please select a valid option.";  }   break; // break the while loop after processing choice once  } } ?>  <!DOCTYPE html> <html> <head>  <title>Login Form - Menu Driven</title> </head> <body>  <h2>Login Form (Menu Driven)</h2>   <form method="post">  <label>Select Operation:</label><br>  <input type="radio" name="choice" value="1"> Login Validation<br>  <input type="radio" name="choice" value="2"> View Credential Rules<br>  <input type="radio" name="choice" value="3"> Exit<br><br>   <div id="loginFields" style="margin-bottom:10px;">  <label>Username:</label>  <input type="text" name="username"><br><br>   <label>Password:</label>  <input type="password" name="password"><br><br>  </div>   <input type="submit" name="submit" value="Submit">  </form>   <?php  if($message != ""){  echo "<h3>$message</h3>";  }  ?>   <script>  // Hide username/password fields unless "Login Validation" is selected  const choiceInputs = document.querySelectorAll('input[name="choice"]');  const loginFields = document.getElementById('loginFields');   choiceInputs.forEach(input => {  input.addEventListener('change', () => {  if(input.value === "1") loginFields.style.display = "block";  else loginFields.style.display = "none";  });  });   // Initially hide fields if no selection  window.onload = () => { loginFields.style.display = "none"; }  </script> </body> </html> |

Output

|  |
| --- |
| Error: Sorry for the inconvenience, there was an API error: Gemini API returned status 429 - {  "error": {  "code": 429,  "message": "You exceeded your current quota, please check your plan and billing details. For more information on this error, head to: https://ai.google.dev/gemini-api/docs/rate-limits.\n\* Quota exceeded for metric: generativelanguage.googleapis.com/generate\_content\_free\_tier\_requests, limit: 50\nPlease retry in 45.094253681s.",  "status": "RESOURCE\_EXHAUSTED",  "details": [  {  "@type": "type.googleapis.com/google.rpc.QuotaFailure",  "violations": [  {  "quotaMetric": "generativelanguage.googleapis.com/generate\_content\_free\_tier\_requests",  "quotaId": "GenerateRequestsPerDayPerProjectPerModel-FreeTier",  "quotaDimensions": {  "model": "gemini-2.0-flash-exp",  "location": "global"  },  "quotaValue": "50"  }  ]  },  {  "@type": "type.googleapis.com/google.rpc.Help",  "links": [  {  "description": "Learn more about Gemini API quotas",  "url": "https://ai.google.dev/gemini-api/docs/rate-limits"  }  ]  },  {  "@type": "type.googleapis.com/google.rpc.RetryInfo",  "retryDelay": "45s"  }  ]  } } |

**Question 6**

Source Code

|  |
| --- |
| <?php /\* Q6. Write a program that passes control to another page using  include, require, exit, and die functions in PHP. \*/  // Message variable $message = "";  // Example usage of include // This will include header.php file if it exists include("Assignment 6 header.php");  // Example usage of require // If footer.php is missing, the script will stop with a fatal error require("Assignment 6 footer.php");  // Simulating some condition $page = "home";  if($page == "about"){  // Redirect control to about.php using include  include("Assignment 6 about.php");  exit(); // stop execution after including about.php } elseif($page == "contact"){  // Redirect control to contact.php using require  require("contact.php");  die("Execution stopped after requiring contact.php");  } else{  $message = "Welcome to the Home Page!"; } ?>  <!DOCTYPE html> <html> <head>  <title>Control Transfer Example</title> </head> <body>  <h2>PHP Control Transfer Example</h2>   <?php  if($message != ""){  echo "<h3>$message</h3>";  }  ?> </body> </html> |

Output

|  |
| --- |
| Error: Sorry for the inconvenience, there was an API error: Gemini API returned status 429 - {  "error": {  "code": 429,  "message": "You exceeded your current quota, please check your plan and billing details. For more information on this error, head to: https://ai.google.dev/gemini-api/docs/rate-limits.\n\* Quota exceeded for metric: generativelanguage.googleapis.com/generate\_content\_free\_tier\_requests, limit: 50\nPlease retry in 42.536241067s.",  "status": "RESOURCE\_EXHAUSTED",  "details": [  {  "@type": "type.googleapis.com/google.rpc.QuotaFailure",  "violations": [  {  "quotaMetric": "generativelanguage.googleapis.com/generate\_content\_free\_tier\_requests",  "quotaId": "GenerateRequestsPerDayPerProjectPerModel-FreeTier",  "quotaDimensions": {  "model": "gemini-2.0-flash-exp",  "location": "global"  },  "quotaValue": "50"  }  ]  },  {  "@type": "type.googleapis.com/google.rpc.Help",  "links": [  {  "description": "Learn more about Gemini API quotas",  "url": "https://ai.google.dev/gemini-api/docs/rate-limits"  }  ]  },  {  "@type": "type.googleapis.com/google.rpc.RetryInfo",  "retryDelay": "42s"  }  ]  } } |

**Question 7**

Source Code

|  |
| --- |
| <?php /\* Q7. Menu-driven Form Validation using filter\_var() in PHP \*/  $message = "";  // Process form data when submitted if(isset($\_POST['submit'])){  $name = $\_POST['name'] ?? "";  $email = $\_POST['email'] ?? "";  $age = $\_POST['age'] ?? "";  $url = $\_POST['url'] ?? "";  $choice = $\_POST['choice'] ?? "";  $continue = true;   while($continue){  switch($choice){  case "1": // Validate Name  if(!filter\_var($name, FILTER\_SANITIZE\_STRING) || !preg\_match("/^[a-zA-Z ]\*$/", $name)){  $message = "Invalid name! Only letters and spaces allowed.";  } else {  $message = "Valid name!";  }  break;   case "2": // Validate Email  if(!filter\_var($email, FILTER\_VALIDATE\_EMAIL)){  $message = "Invalid email format!";  } else {  $message = "Valid email!";  }  break;   case "3": // Validate Age  if(!filter\_var($age, FILTER\_VALIDATE\_INT, array("options"=>array("min\_range"=>1, "max\_range"=>120)))){  $message = "Invalid age! Must be a number between 1 and 120.";  } else {  $message = "Valid age!";  }  break;   case "4": // Validate URL  if(!filter\_var($url, FILTER\_VALIDATE\_URL)){  $message = "Invalid URL!";  } else {  $message = "Valid URL!";  }  break;   case "5": // Validate All  $errors = "";  if(!filter\_var($name, FILTER\_SANITIZE\_STRING) || !preg\_match("/^[a-zA-Z ]\*$/", $name)){  $errors .= "Invalid name! Only letters and spaces allowed.<br>";  }  if(!filter\_var($email, FILTER\_VALIDATE\_EMAIL)){  $errors .= "Invalid email format!<br>";  }  if(!filter\_var($age, FILTER\_VALIDATE\_INT, array("options"=>array("min\_range"=>1, "max\_range"=>120)))){  $errors .= "Invalid age! Must be 1-120.<br>";  }  if(!filter\_var($url, FILTER\_VALIDATE\_URL)){  $errors .= "Invalid URL!<br>";  }  $message = $errors == "" ? "All inputs are valid. Form submitted successfully!" : $errors;  break;   case "6": // Exit  $continue = false;  $message = "Exited menu.";  break;   default:  $message = "Please select a valid option.";  }   break; // Break after one iteration  } } ?>  <!DOCTYPE html> <html> <head>  <title>Filter\_var Form Validation - Menu Driven</title> </head> <body>  <h2>Form Validation Example (Menu Driven)</h2>   <form method="post">  <label>Select Validation Operation:</label><br>  <input type="radio" name="choice" value="1"> Validate Name<br>  <input type="radio" name="choice" value="2"> Validate Email<br>  <input type="radio" name="choice" value="3"> Validate Age<br>  <input type="radio" name="choice" value="4"> Validate URL<br>  <input type="radio" name="choice" value="5"> Validate All Fields<br>  <input type="radio" name="choice" value="6"> Exit<br><br>   <div id="formFields" style="margin-bottom:10px;">  <label>Name:</label>  <input type="text" name="name"><br><br>   <label>Email:</label>  <input type="text" name="email"><br><br>   <label>Age:</label>  <input type="number" name="age"><br><br>   <label>Website URL:</label>  <input type="text" name="url"><br><br>  </div>   <input type="submit" name="submit" value="Submit">  </form>   <?php  if($message != ""){  echo "<h3>$message</h3>";  }  ?>   <script>  // Show/hide form fields based on choice  const choiceInputs = document.querySelectorAll('input[name="choice"]');  const formFields = document.getElementById('formFields');   choiceInputs.forEach(input => {  input.addEventListener('change', () => {  if(input.value !== "6") formFields.style.display = "block";  else formFields.style.display = "none";  });  });   window.onload = () => { formFields.style.display = "none"; }  </script> </body> </html> |

Output

|  |
| --- |
| Error: Sorry for the inconvenience, there was an API error: Gemini API returned status 429 - {  "error": {  "code": 429,  "message": "You exceeded your current quota, please check your plan and billing details. For more information on this error, head to: https://ai.google.dev/gemini-api/docs/rate-limits.\n\* Quota exceeded for metric: generativelanguage.googleapis.com/generate\_content\_free\_tier\_requests, limit: 50\nPlease retry in 39.878486423s.",  "status": "RESOURCE\_EXHAUSTED",  "details": [  {  "@type": "type.googleapis.com/google.rpc.QuotaFailure",  "violations": [  {  "quotaMetric": "generativelanguage.googleapis.com/generate\_content\_free\_tier\_requests",  "quotaId": "GenerateRequestsPerDayPerProjectPerModel-FreeTier",  "quotaDimensions": {  "model": "gemini-2.0-flash-exp",  "location": "global"  },  "quotaValue": "50"  }  ]  },  {  "@type": "type.googleapis.com/google.rpc.Help",  "links": [  {  "description": "Learn more about Gemini API quotas",  "url": "https://ai.google.dev/gemini-api/docs/rate-limits"  }  ]  },  {  "@type": "type.googleapis.com/google.rpc.RetryInfo",  "retryDelay": "39s"  }  ]  } } |

**Question 8**

Source Code

|  |
| --- |
| <?php /\* Q8. Menu-driven Cookie Example in PHP \*/  $message = "";  if(isset($\_POST['submit'])){  $choice = $\_POST['choice'] ?? "";  $continue = true;   while($continue){  switch($choice){  case "1": // Set Cookie  setcookie("username", "Yourname", time() + 86400, "/"); // expires in 24 hours  $message = "Cookie 'username' has been set. Refresh the page to see the welcome message.";  break;   case "2": // Check Cookie  if(isset($\_COOKIE['username'])){  $message = "Welcome back, " . $\_COOKIE['username'] . "!";  } else {  $message = "Hello, new visitor! Cookie not found. You may set it first.";  }  break;   case "3": // Delete Cookie  setcookie("username", "", time() - 3600, "/"); // expire cookie  $message = "Cookie 'username' has been deleted.";  break;   case "4": // Exit  $continue = false;  $message = "Exited menu.";  break;   default:  $message = "Please select a valid option.";  }   break; // Break after one iteration  } } ?>  <!DOCTYPE html> <html> <head>  <title>Cookie Example - Menu Driven</title> </head> <body>  <h2>PHP Cookie Example (Menu Driven)</h2>   <form method="post">  <label>Select Operation:</label><br>  <input type="radio" name="choice" value="1"> Set Cookie<br>  <input type="radio" name="choice" value="2"> Check Cookie<br>  <input type="radio" name="choice" value="3"> Delete Cookie<br>  <input type="radio" name="choice" value="4"> Exit<br><br>   <input type="submit" name="submit" value="Submit">  </form>   <?php  if($message != ""){  echo "<h3>$message</h3>";  }  ?> </body> </html> |

Output

|  |
| --- |
| {'usageMetadata': {'promptTokenCount': 849, 'totalTokenCount': 849, 'promptTokensDetails': [{'modality': 'TEXT', 'tokenCount': 849}]}, 'modelVersion': 'gemini-2.0-flash-exp', 'responseId': '5jjZaI\_WMZvnjMcPxc7mmAM'} |

**Question 9**

Source Code

|  |
| --- |
| <?php /\* Q9. Menu-driven PHP Session Visit Counter \*/  session\_start();  $message = "";  // Initialize visit count if not set if(!isset($\_SESSION['visits'])){  $\_SESSION['visits'] = 0; }  if(isset($\_POST['submit'])){  $choice = $\_POST['choice'] ?? "";  $continue = true;   while($continue){  switch($choice){  case "1": // View Visit Count  $\_SESSION['visits']++; // Increment visit count  $message = "You have visited this page <b>" . $\_SESSION['visits'] . "</b> times in this session.";  break;   case "2": // Reset Visit Count  $\_SESSION['visits'] = 0;  $message = "Visit count has been reset.";  break;   case "3": // Exit  $continue = false;  $message = "Exited menu.";  break;   default:  $message = "Please select a valid option.";  }   break; // Break after one iteration  } } ?>  <!DOCTYPE html> <html> <head>  <title>Session Visit Counter - Menu Driven</title> </head> <body>  <h2>PHP Session Visit Counter (Menu Driven)</h2>   <form method="post">  <label>Select Operation:</label><br>  <input type="radio" name="choice" value="1"> View Visit Count<br>  <input type="radio" name="choice" value="2"> Reset Visit Count<br>  <input type="radio" name="choice" value="3"> Exit<br><br>   <input type="submit" name="submit" value="Submit">  </form>   <?php  if($message != ""){  echo "<h3>$message</h3>";  }  ?> </body> </html> |

Output

|  |
| --- |
| {'usageMetadata': {'promptTokenCount': 772, 'totalTokenCount': 772, 'promptTokensDetails': [{'modality': 'TEXT', 'tokenCount': 772}]}, 'modelVersion': 'gemini-2.0-flash-exp', 'responseId': '6jjZaMLVAeKSjMcPsIHjqAM'} |