Questions and Answers

1. What is server-side scripting? What are the advantages of server-side scripting?  
   🡪Server-side scripting is when code runs on the web server to create webpages. The server processes requests, like fetching data, and sends the result to the browser.

Advantages:

* Secure: Code isn’t visible to users.
* Database access: Can store or get data.
* Custom content: Shows different pages for each user.
* Faster browsing: Reduces work for the user’s device.

1. What is PHP? Explain the history of PHP.  
   🡪PHP is a free programming language for making interactive websites. It’s called Hypertext Preprocessor and works well with HTML and databases.  
   History:

* Created in 1994 by Rasmus Lerdorf for his website.
* First named Personal Home Page Tools.
* Grew into a powerful language for web tasks.
* Today, used by many websites like forums and shops.

1. What is a variable? Write down the different rules to create a variable in PHP.  
   🡪A variable stores data, like a number or word, for use in your code.  
   Rules:

* Starts with $.
* First character after $ is a letter or underscore.
* No numbers at the start.
* Only letters, numbers, or underscores allowed.  
    
  $name = "Sara";  
  $age = 20;

1. What is a super global variable in PHP? Explain the different types of super global variables in PHP.  
   🡪Super global variables are special PHP variables you can use anywhere in your code. They hold data like user inputs or server info.

Types:

* $\_GET: Data from URLs.
* $\_POST: Data from forms.
* $\_REQUEST: Mix of $\_GET, $\_POST, and cookies.
* $\_SESSION: User data for their visit.
* $\_COOKIE: Data stored in the browser.
* $\_FILES: Uploaded files.
* $\_SERVER: Server details.
* $\_ENV: System settings.
* $GLOBALS: All global variables.

1. Describe different PHP operators in PHP. Explain ternary operators.  
   🡪Operators are symbols for math, comparisons, or logic.  
   Types:

* Arithmetic: + (add), - (subtract), \* (multiply), / (divide), % (remainder).
* Assignment: = (set), += (add and set), -= (subtract and set).
* Comparison: == (equal), === (exactly equal), != (not equal), <, >.
* Logical: && (and), || (or), ! (not).
* String: . (join strings), .= (add to string).
* Array: + (merge), == (compare).  
  Ternary operator: A short if-else. Checks a condition and picks a value.  
    
  $age = 16;  
  $status = ($age >= 18) ? "Adult" : "Minor";  
  echo $status;

1. Describe different control structures in PHP.  
   🡪Control structures choose or repeat code based on conditions.  
   Types:

* if/else/elseif: Runs code if true, else runs other code.
* switch: Selects code based on a value.
* for: Loops a set number of times.
* while: Loops while true.
* do...while: Loops at least once, then checks.
* foreach: Loops through arrays.

1. What is the use of foreach loop in PHP? Explain using examples.  
   🡪The foreach loop goes through each item in an array, like reading a list.  
     
   $fruits = array("Apple", "Banana", "Orange");  
   foreach ($fruits as $fruit) {  
   echo $fruit . " ";  
   }
2. What are functions in PHP? How to use functions in PHP?  
   🡪Functions are reusable code blocks you call by name. They save time by avoiding repeated code.  
     
   function sayHello($name) {  
   return "Hi, $name!";  
   }  
   echo sayHello("Tom");
3. What is an array? What are the different types of arrays in PHP? Explain with the help of examples.  
   🡪An array holds multiple values in one variable, like a list.  
   Types:

* Indexed: Uses numbers (0, 1, 2...) as keys.  
    
  $days = array("Monday", "Tuesday", "Wednesday");  
  echo $days[0];
* Associative: Uses words as keys.  
    
  $scores = array("math" => 85, "english" => 90);  
  echo $scores["math"];
* Multidimensional: Arrays inside arrays.  
    
  $classes = array(  
  array("Ali", 95),  
  array("Zara", 88)  
  );  
  echo $classes[1][1];

1. Define function overloading. How is function overloading handled in PHP? Explain with an example.  
   🡪Function overloading is having multiple functions with the same name but different inputs. PHP doesn’t allow this directly but uses \_\_call() to handle it.  
     
   class Test {  
   function \_\_call($name, $args) {  
   echo "$name called with: ";  
   print\_r($args);  
   }  
   }  
   $obj = new Test();  
   $obj->run(1, "test"); )
2. Explain the different array methods/functions used in PHP.  
   🡪Array functions help manage lists:

* count(): Counts items.
* array\_push(): Adds to the end.
* array\_pop(): Removes the last item.
* array\_merge(): Joins arrays.
* sort(): Sorts in order.
* array\_reverse(): Flips the order.

1. Describe Form Handling in PHP and Form validation in PHP.  
   🡪Form handling collects user inputs from forms using $\_GET or $\_POST. Validation checks if inputs are correct, like not empty.  
     
   if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
   $email = $\_POST["email"];  
   if (empty($email)) {  
   echo "Email is required";  
   } else {  
   echo "Got email: $email";  
   }  
   }
2. Describe the use case of time function in PHP.  
   🡪The time() function returns the current time as a number (timestamp). It’s used for things like logging when a user visits.  
     
   echo time();  
   echo date("Y-m-d", time());
3. Describe file handling in PHP.  
   🡪File handling lets PHP read or write files, like saving user notes.  
   Functions:

* fopen(): Opens a file.
* fread(): Reads content.
* fwrite(): Writes content.
* fclose(): Closes the file.  
    
  $file = fopen("note.txt", "w");  
  fwrite($file, "Welcome");  
  fclose($file);

1. What are the different types of error in PHP? Explain how error handling is done in PHP.  
   🡪Errors are mistakes in code. Types:

* Parse: Bad syntax, stops everything.
* Fatal: Big error, like missing function, stops code.
* Warning: Issue but code runs, like a missing file.
* Notice: Small issue, like unset variable.  
  Error handling: Use try-catch or custom functions.  
    
  set\_error\_handler("myError");  
  function myError($num, $msg) {  
  echo "Error: $msg";  
  }  
  echo $noVar;

1. How file is uploaded in the server using PHP? Explain in detail using an example.  
   🡪File uploading lets users send files to the server, like images. You need a form and PHP to move the file.

<form method="post" enctype="multipart/form-data">

<input type="file" name="myfile">

<input type="submit" name="submit">

</form>

<?php if (isset($\_POST["submit"]))

{ $target = "uploads/" . $\_FILES["myfile"]["name"]; move\_uploaded\_file($\_FILES["myfile"]["tmp\_name"], $target);

echo "File uploaded";

} ?>

1. What is a session? How does session work?  
   🡪A session saves user data on the server during their visit, like their name. Start it with session\_start().  
     
   session\_start();  
   $\_SESSION["user"] = "Lila";  
   echo $\_SESSION["user"];
2. Describe the different session functions used in PHP.  
   🡪Session functions manage user data:

* session\_start(): Starts a session.
* $\_SESSION[]: Sets or gets data.
* session\_destroy(): Ends a session.
* session\_unset(): Clears session data.

1. How can you connect to a database using PHP? Give examples to support your answer.  
   🡪PHP connects to databases like MySQL to store or fetch data.  
     
   $conn = mysqli\_connect("localhost", "root", "", "shop");  
   if (!$conn) {  
   die("Failed to connect");  
   }  
   echo "Connected";
2. What are the different SQL data types?  
   🡪SQL data types say what a column holds:

* INT: Numbers like 5.
* VARCHAR: Short text like "Bob".
* TEXT: Long text like notes.
* DATE: Dates like 2025-04-10.
* DATETIME: Date and time like 2025-04-10 15:00:00.
* FLOAT: Decimals like 2.99.
* BOOLEAN: True or false.

1. With a suitable example each, describe:  
   a. Data insertion using PHP:

🡪 Adds new data.  
  
$sql = "INSERT INTO users (name) VALUES ('Emma')";  
mysqli\_query($conn, $sql);

b. Data Selection using PHP: Gets data.  
  
$sql = "SELECT name FROM users";  
$result = mysqli\_query($conn, $sql);

c. Data Update using PHP: Changes data.  
  
$sql = "UPDATE users SET name='Liam' WHERE id=1";  
mysqli\_query($conn, $sql);

d. Data delete using PHP: Removes data.  
  
$sql = "DELETE FROM users WHERE id=1";  
mysqli\_query($conn, $sql);

1. What are aggregate functions in MySQL?  
   🡪Aggregate functions calculate values across rows:

* COUNT(): Counts rows.
* SUM(): Adds numbers.
* AVG(): Finds average.
* MIN(): Gets smallest value.
* MAX(): Gets largest value.

1. Describe subqueries and joins using examples.

* Subquery: A query inside another to find specific data.  
    
  $sql = "SELECT name FROM users WHERE id = (SELECT user\_id FROM orders WHERE order\_id = 1)";
* Join: Links tables to combine data.  
    
  $sql = "SELECT users.name, orders.total FROM users INNER JOIN orders ON users.id = orders.user\_id";

1. Difference between:  
   a) get and post:

* GET: Data in URL, less secure, for searches.
* POST: Data hidden, more secure, for forms.
* b) time and strtotime:
* time(): Current time as a number.
* strtotime(): Turns text like "tomorrow" into a number.
* c) order by and group by:
* ORDER BY: Sorts rows.
* GROUP BY: Groups rows for calculations.
* d) include and require:
* include: Warns if file missing, continues.
* require: Stops if file missing.
* e) include\_once and require\_once:
* Both load a file only once.
* f) echo and print:
* echo: Shows text, fast.
* print: Shows text, returns 1.
* g) cookie and session:
* Cookie: Stored in browser, lasts longer.
* Session: Stored on server, ends after visit.