

# LARAVEL DEVELOPER

## HANDBOOK 2025



```
</div>
<!-- /Main Nav -->

<!-- Aside Nav -->
<div id="aside">
  <ul class="list-group">
    <li><a href="#">Home</a></li>
    <li><a href="#">Categories</a></li>
    <li><a href="#">About Us</a></li>
    <li><a href="#">Contacts</a></li>
    <li><a href="#">Advertise</a></li>
  </ul>
  <button class="nav-close nav-aside-close"><span></span></button>
</div>
<!-- /Aside Nav -->

</div>
<!-- /NAV -->
```

অনলাইন কিংবা অফলাইন। এখনকার চাকরির বাজার কেটো প্রতিযোগিতামূলক তা আশা করি আর বলে দিতে হবে না। শত শত চাকরির প্রস্তাব, কিন্তু হাজার হাজার প্রতিযোগী! এই অস্থিতিশীল চাকরির বাজারে টিকতে হল ভালোভাবে প্রস্তুতি নেয়া খুবই গুরুত্বপূর্ণ, সর্ঠিক দিক-নির্দেশনা এবং আত্মবিশ্বাস সহ আরো অনেক কিছুর। এ যেন যুক্তির চেয়েও কিছু অংশে কম নয়!

আর তাই আপনাকে একজন ডেভেলপার হিসেবে জব ইন্টারভিউ নামক যুক্তির মধ্যান্তের জন্য রেডি করার জন্যই “**Laravel Developer Handbook 2025**” এই ইন্টারভিউ হ্যান্ডবুক।

আমরা আশা রাখছি, হ্যান্ডবুকের প্রতিটি অধ্যায় আপনাকে আপনার পরবর্তী ইন্টারভিউয়ের জন্য পর্যাপ্ত সাহস, আত্মবিশ্বাস এবং প্রেমার নলেজ দিবে যা আপনাকে যথাযথভাবে প্রস্তুত হতে হল্ল করবে; যেনো ইন্টারভিউ বোর্ডে অন্য দশজনের চেয়ে আপনাকে বিশেষ ভাবে আলাদা করা যেতে পারে।

## এই হ্যান্ডবুকের মূল উদ্দেশ্যঃ

এই হ্যান্ডবুকের মাধ্যমে আমরা আপনাকে দরকারি তথ্য দিতে চাই, পাশাপাশি আত্মবিশ্বাস গড়তে সাহায্য করব, যেন আপনি সহজেই আপনার স্বন্মের চাকরির জন্য নিজেকে প্রস্তুত করতে পারেন। তাই এখানে শুধু টেকনিক্যাল প্রশ্ন নয়, বরং গুরুত্বপূর্ণ সফট স্কিল, ইন্টারভিউ বোর্ডে আসা বিভিন্ন ধরনের প্রশ্ন ও আচরণগত বিষয়গুলোও অন্তর্ভুক্ত করার চেষ্টা করেছি।

আপনি যদি প্রথম চাকরির জন্য প্রস্তুতি নিচ্ছেন, বা ইতিমধ্যে কাজ করছেন কিন্তু নতুন কোনো জব খুঁজছেন- যেটাই হোক, আমরা সবসময় আপনাকে সামোর্ট দিতে প্রস্তুত!

তবে একটা বিষয় মাথায় রাখা জরুরি- এই হ্যান্ডবুক কেবল সহায়ক। সম্পূর্ণ প্রস্তুতির জন্য আপনাকে আরও বিভিন্ন আর্টিকেল, ডকুমেন্টেশন, ব্লগ, ভিডিও ইত্যাদি থেকে শিখতে হবে।

আপনার জন্য শুভকামনা! নতুন অভিজ্ঞতার জন্য অগ্রিম শুভেচ্ছা!



# সুচিপত্র:

- ▶ মুখ্যবন্ধ
- ▶ সফট স্কিল ভিত্তিক প্রশ্ন
- ▶ ইলেক্ট্রনিক প্রস্তুতি এবং কিছু পরামর্শ
- ▶ টেকনিক্যাল স্কিল ভিত্তিক প্রশ্ন
- ▶ Technical PHP Questions
- ▶ Technical Laravel Questions
- ▶ আরেকটুখানি বিসের্স
- ▶ শেষ কথা

## টেকনিক্যাল স্কিল ভিত্তিক প্রশ্ন

চাকরির ইন্টারভিউয়ে বিশেষ করে ডেভেলপার কিংবা প্রোগ্রামারদের অন্যান্য প্রশ্নাবলীর পাশাপাশি টেকনিক্যাল প্রশ্নাওরের মাধ্যমে যোগ্যতা যাচাই করা হয়ে থাকে। চলুন এবার টেকনিক্যাল স্কিল সম্পর্কিত তাত্ত্বিক বিষয় ঝালিয়ে নেই।

পরবর্তী বেশ কয়েকটি পরিচ্ছেদে আমরা প্রথমে PHP এর কিছু টেকনিক্যাল প্রশ্নাবলী দেখব। এরপরে লারাভেলের বেশ কিছু প্রশ্নাবলীর মাধ্যমে ইন্টারভিউয়ের পূর্বের মূহূর্ত পর্যন্ত নিজের জ্ঞান-অজ্ঞানকে ঝালিয়ে নেয়ার মাধ্যমে এই অধ্যায়টি শেষ করবো। তবে লক্ষ্য করলে দেখবেন যে, প্রতিটি পরিচ্ছেদেই Laravel Career Path এর আউটলাইনের সাথে কিছু না কিছু মিল খুঁজে পাবেন।

তবে আগেভাগেই জানিয়ে রাখছি টেকনিক্যাল অনেকগুলো বিষয়, অনেকগুলো শব্দ রয়েছে যেগুলোর সরাসরি বাংলা অর্থ প্রকৃত মর্মার্থ প্রকাশ করার বদলে সেটাকে আরো ঘোলাটে করে দেয়। সেজন্য আমরা পরবর্তী সম্পূর্ণ টেকনিক্যাল স্কিল ভিত্তিক প্রশ্নাবলীর অধ্যায়টিকে ইংরেজি ভাষায় রেখেছি।

ইংরেজিতে হ্যাত কিছুটা দুর্বলতা রয়েছে দেখে মন খারাপ করছেন, তাই না? চিন্তা করবেন না, যথাসম্ভব সহজবোধ্য ইংরেজি শব্দ ব্যবহার করে প্রশ্ন এবং উত্তর গুলোকে বোধগম্য রাখার চেষ্টা করা হয়েছে। তাই সাহস হারাবেন না আশা করছি। চলুন শুরু করা যাক!



# Technical PHP Questions

## Introduction to PHP

### 1. Describe the key features of PHP that make it suitable for web development.

**Answer:** PHP supports various databases, is platform-independent, open-source, and has a large community. It's easy to learn, has extensive documentation, and offers support for different web servers. These features make PHP a popular choice for web development.

### 2. Explain the difference between client-side and server-side scripting.

**Answer:** Client-side scripting occurs on the user's browser, while server-side scripting happens on the web server. PHP is a server-side scripting language, meaning it executes on the server, processes data, and then sends the result to the client's browser.

### 3. Mention some popular websites or projects that use PHP.

**Answer:** PHP is used in many well-known websites like WordPress, Facebook, Wikipedia, and Joomla. It's also a key technology in various content management systems and e-commerce platforms.



## Why and How PHP was Created

**1. What was the original purpose behind creating PHP, and who is its creator?**

**Answer:** PHP was initially created by Rasmus Lerdorf in 1994 to track visits to his online resume. It evolved into a server-side scripting language to aid in web development.

**2. Explain the evolution of PHP from a set of scripts to a full-fledged programming language.**

**Answer:** PHP started as a set of Common Gateway Interface (CGI) binaries written in C. Over time, it evolved with the addition of features like functions, database connectivity, and object-oriented programming, transforming into a comprehensive programming language.

**3. Discuss the significance of PHP's open-source nature in its development.**

**Answer:** PHP being open-source encourages collaboration and community contributions. This openness has led to continuous improvements, bug fixes, and the development of various extensions and frameworks by a large community of developers worldwide.



#### 4. How has the PHP community contributed to the language's growth and development?

**Answer:** The PHP community actively contributes to PHP's growth through bug reporting, proposing new features, and developing extensions and frameworks. Community-driven efforts, including PHP conferences and forums, play a crucial role in shaping the language's future.

#### 5. In what ways has PHP's focus on web development influenced its design and feature set?

**Answer:** PHP's design has been influenced by its focus on web development through features like easy integration with HTML, support for server-side scripting, & extensive libraries for web-related tasks. Its evolution is closely tied to the demands of web development.

### PHP CLI vs PHP FPM:

#### 1. Differentiate between PHP CLI and PHP FPM in terms of their use cases.

**Answer:** PHP CLI (Command Line Interface) is used for running PHP scripts from the command line, while PHP FPM (FastCGI Process Manager) is used for serving PHP in a FastCGI environment, typically for web servers. CLI is suitable for scripts, and FPM is for web server integration.



## 2. Explain the primary advantages of using PHP CLI for script execution.

**Answer:** TPHP CLI provides a convenient way to run PHP scripts directly from the command line, making it suitable for tasks like batch processing, automation, and cron jobs. It allows developers to test scripts without the need for a web server.

## 3. Discuss the role of PHP FPM in enhancing the performance of web applications.

**Answer:** PHP FPM improves web application performance by managing multiple PHP processes, optimizing resource usage, and handling requests efficiently. It is especially beneficial in a high-traffic web environment.

## 4. When might you choose PHP CLI over PHP FPM, and vice versa?

**Answer:** Choose PHP CLI for script execution and tasks not related to web serving, such as batch processing. PHP FPM is suitable for serving PHP in a web environment, providing the necessary features for handling HTTP requests efficiently.

## 5. How does PHP FPM contribute to the scalability of web applications?

**Answer:** PHP FPM enhances scalability by efficiently managing PHP processes, allowing web servers to handle a large number of concurrent requests. It helps distribute the workload effectively, contributing to improved performance in high-traffic scenarios.



## Data Types

### 1. What are data types in PHP?

**Answer:** Data types in PHP define the type of data a variable can hold, such as integer, string, float, boolean, etc.

### 2. Can you give examples of scalar data types in PHP?

**Answer:** Scalar data types in PHP include int, float, string, and bool. For example, \$age = 21; \$price = 12.99; \$name = "Nahian"; \$isStudent = true;

### 3. Explain the difference between integers and floats.

**Answer:** Integers are whole numbers without decimal points, like 5 or -10. Floats (or doubles) can have decimal points, like 3.14 or -7.5.

### 4. How do you check the data type of a variable in PHP?

**Answer:** You can use the `gettype($variable)` function to check the data type of a variable. For example, `gettype($age)` would return "integer."

### 5. Discuss the importance of understanding data types in PHP programming.

**Answer:** Understanding data types is crucial for proper variable usage and preventing unexpected behavior. It ensures accurate calculations, comparisons, and data manipulation in PHP.



## Type Hinting

### 1. What is type hinting in PHP?

**Answer:** Type hinting is a feature that allows you to specify the expected data type of a function's parameter or return value.

### 2. When is type hinting commonly used?

**Answer:** Type hinting is commonly used in function declarations to specify the data type a function expects for its parameters or return value.

### 3. Can you use type hinting for scalar data types like strings or booleans?

**Answer:** Yes, type hinting for scalar data types (string, bool, int, float) is allowed since PHP 7. PHP has supported function parameter type hinting from version 5.0, but only for objects, and from version 5.1 for arrays as well. With PHP 7, scalar types can be type-hinted as well, making it one of the more exciting features of the release.

### 4. How does type hinting contribute to code clarity and robustness?

**Answer:** Type hinting improves code clarity by explicitly stating the expected data types, reducing errors. It also enhances code robustness by preventing unintended data type mismatches.



## Type Casting in PHP

### 1. What is type casting in PHP?

**Answer:** Type casting in PHP is the process of converting one data type into another. It helps in manipulating data by changing its type to perform specific operations.

### 2. Explain the difference between implicit and explicit type casting.

**Answer:** Implicit type casting is done automatically by PHP, while explicit type casting requires manual intervention. Implicit happens automatically during operations, whereas explicit involves using functions like (int), (float), (string), etc., to cast data types.

### 3. How is it possible to cast types in PHP?

**Answer:** The name of the output type has to be specified in parentheses before the variable which is to be cast as follows:

- (int), (integer) – cast to integer
- (bool), (boolean) – cast to boolean
- (float), (double), (real) – cast to float
- (string) – cast to string
- (array) – cast to array
- (object) – cast to object



#### 4. Discuss the difference between (int) and intval() for type casting.

**Answer:** Both (int) and intval() can be used to convert a value to an integer. However, (int) is a language construct, and intval() is a function. The primary difference is that intval() provides more flexibility as it accepts a base parameter for hexadecimal or octal conversion.

#### 5. Explain the potential issues with type juggling in PHP.

**Answer:** Type juggling in PHP can lead to unexpected results if not handled carefully. For example, when comparing two values of different types using loose equality (==), PHP may perform type conversion. This can lead to unintended outcomes, and it's generally recommended to use strict comparison (===) to avoid such issues.

## Type Casting in PHP

### 1. What is a variable in PHP?

**Answer:** A variable in PHP is a container for storing data. It has a name, a data type, and a value.

### 2. How do you declare a variable in PHP?

**Answer:** You can declare a variable in PHP using the dollar sign (\$) followed by the variable name. For example: \$name = "Nahian";



### 3. Explain the rules for naming variables in PHP.

**Answer:** Variable names in PHP must start with a letter or underscore, followed by letters, numbers, or underscores. They are case-sensitive and should not be PHP reserved words.

### 4. Can you change the value of a variable after it has been declared?

**Answer:** Yes, you can change the value of a variable after it has been declared by assigning a new value to it.

### 5. Why are variables essential in PHP programming?

**Answer:** Variables are essential in PHP programming because they allow the storage and manipulation of data.

## Arrays

### 1. What is an array in PHP?

**Answer:** An array in PHP is a data structure that stores multiple values under a single variable name. It allows you to access these values using an index.

### 2. How do you declare an array in PHP?

**Answer:** You can declare an array in PHP using the `array()` construct or the short syntax `[]`. For example: `$colors = array("red", "green", "blue");` or `$names = ["nahian", "bruce", "wayne"];`



### 3. Explain the concept of array indices or indexes.

**Answer:** Array indices or indexes are numeric or string keys that identify elements in an array. They start from zero for the first element and increase sequentially.

### 4. Can an array in PHP store values of different data types?

**Answer:** Yes, an array in PHP can store values of different data types, including integers, strings, booleans, and even other arrays.

### 5. How do you access a specific element in an array?

**Answer:** You can access a specific element in an array using its index within square brackets. For example: \$colors[1] would return "green" in the previous example. They make code more flexible, readable, and dynamic.

## Array Manipulation

### 1. What is array manipulation in PHP?

**Answer:** Array manipulation in PHP refers to performing operations on arrays, such as adding or removing elements, sorting, merging, and other modifications.



## 2. How can you add a new element to the end of an array in PHP?

**Answer:** You can use the [ ] notation or the array\_push() function to add a new element to the end of an array.

## 3. Explain the difference between array\_merge() and the + operator for merging arrays.

**Answer:** array\_merge() merges arrays and creates a new one, while the + operator preserves the keys of the first array and only adds the unique keys from the second array.

## 4. How do you remove an element from an associative array based on its key?

**Answer:** You can use the unset(\$array['key']) function to remove an element from an associative array based on its key.

## 5. Discuss the significance of array manipulation in PHP programming.

**Answer:** Array manipulation is crucial in PHP programming for dynamic data handling. It allows developers to modify, organize, and extract data efficiently, enhancing the flexibility of their applications.



## Functions

### 1. What is a function in PHP?

**Answer:** A function in PHP is a block of reusable code that performs a specific task. It is defined using the `function` keyword.

### 2. How do you call a function in PHP?

**Answer:** You can call a function in PHP by using its name followed by parentheses. For example: `myFunction()`.

### 3. What is the purpose of parameters in a function?

**Answer:** Parameters in a function allow you to pass values to the function, making it more flexible and capable of handling different inputs.

### 4. Can a function in PHP return a value?

**Answer:** Yes, a function in PHP can return a value using the `return` keyword. The returned value can be used in other parts of the code.

### 5. Why is code modularization important, and how do functions contribute to it?

**Answer:** Code modularization is essential for better organization and maintainability. Functions contribute by encapsulating specific functionality, making code more readable, reusable, and easier to maintain.



## Built-In Functions

### 1. What are built-in functions in PHP?

**Answer:** Built-in functions in PHP are pre-defined functions provided by the language to perform common tasks, such as string manipulation, array operations, and mathematical calculations.

### 2. What is the function `file_get_contents()` useful for?

**Answer:** `file_get_contents()` lets reading a file and storing it in a string variable.

### 3. Explain the purpose of the `date()` function in PHP.

**Answer:** The `date()` function is used to format and display the current date and time in PHP.

### 4. Why are built-in functions beneficial in PHP programming?

**Answer:** Built-in functions save time and effort by providing ready-made solutions for common tasks. They enhance code readability and maintainability by promoting a standardized approach to certain operations.

```
<li><a href="contact.html">Contact</a>
<li><a href="#">Advertiser</a>
</ul>
<button class="nav-close nav">Close</button>
</div>
<!-- /Aside Nav --&gt;
&lt;/div&gt;
&lt;!-- /NAV --&gt;</pre>
```



## 5. Can you explain the use of the header() function in PHP and provide examples of its different applications?

**Answer:** The header() function in PHP serves multiple purposes: facilitating page redirection (e.g., header("Location: index.php")), sending HTTP status codes (e.g., header("HTTP/1.0 404 Not Found")), and delivering raw HTTP headers (e.g., header('Content-Type: application/json')). These functionalities enable tasks such as navigation, error handling, and specifying content types for dynamic content or APIs.

## Function Named Arguments

### 1. What are named arguments in PHP functions?

**Answer:** Named arguments in PHP functions allow you to pass values to a function by explicitly mentioning the parameter names, improving code clarity.

### 2: Discuss the advantages of using named arguments in PHP.

**Answer:** Named arguments improve code readability, make it clear which value corresponds to which parameter, and allow flexibility in changing the order of arguments without affecting the function call.



### 3: Can you mix named and traditional positional arguments in a function call?

**Answer:** Yes, you can mix named and positional arguments, but named arguments must come after positional ones in the function call.

### 4: How do named arguments contribute to code maintenance and collaboration?

**Answer:** Named arguments enhance code maintenance by making it clear which values are associated with which parameters. This clarity improves collaboration among developers, especially in larger codebases.

## Closures

### 1. What is a closure in PHP?

**Answer:** A closure in PHP is a function without a name, defined using the function keyword. It can capture and use variables from its surrounding scope.

### 2. Explain the concept of capturing variables in closures.

**Answer:** Closures can capture and use variables from the surrounding scope, even after the parent function has finished executing. This is known as variable capture.

### 3. Can closures be passed as arguments to other functions?

**Answer:** Yes, closures can be passed as arguments to other functions, allowing for more dynamic and flexible code.

### 4. Discuss the practical applications of closures in PHP programming.

**Answer:** Closures are useful for creating more concise and readable code, especially in scenarios like callback functions, event handling, and situations where a short-lived function is needed.

## Loops

### 1. What is a loop in PHP, and why is it used?

**Answer:** A loop in PHP is a control structure that allows you to repeat a block of code multiple times. It is used for tasks that require repetitive execution.

### 2. How does a for loop work in PHP?

**Answer:** A for loop in PHP has three parts: initialization, condition, and increment. It repeatedly executes a block of code as long as the condition is true.



### 3. Explain the purpose of a foreach loop in PHP.

**Answer:** A foreach loop in PHP is used to iterate over elements in an array or other iterable objects. It simplifies the process of iterating through the entire array.

### 4. What is an infinite loop, and how can it be avoided?

**Answer:** An infinite loop is a loop that never stops executing. It can be avoided by ensuring that the loop condition becomes false at some point during execution.

### 5. Discuss the advantages of using loops for repetitive tasks in PHP.

**Answer:** Loops in PHP make code more efficient and concise for tasks that involve repetitive actions, such as processing array elements, printing patterns, or performing calculations iteratively. They reduce redundancy and enhance code readability.





FULL STACK  
**LARAVEL**  
**Career Path**

with **PHP** | **VUE.JS** | **AI**

ক্যারিয়ার পাথে এন্ডোল করতে স্ক্যান করুন



```
</ul>
<button class="nav-close nav">
</div>
<!-- /Aside Nav -->
</div>
<!-- /NAV -->
```