Python Full Stack

Basic PHP Full Stack Interview Questions (1-25)

1. **What is PHP?**
    - PHP (Hypertext Preprocessor) is a server-side scripting language used to create dynamic web pages. It can interact with databases, and handle form submissions, and session management.
2. **What are the different data types in PHP?**
    - PHP supports several data types: Integer, Float, String, Boolean, Array, Object, and NULL.
3. **What is the difference between `echo` and `print` in PHP?**
    - `echo` can output one or more strings, while `print` can only output one string and always returns 1.
4. **How do you define a constant in PHP?**
    - Constants in PHP are defined using the `define()` function. For example: `define('CONSTANT_NAME', 'value');`.
5. **What is the purpose of the `isset()` function in PHP?**
    - `isset()` checks if a variable is set and is not `NULL`. It returns `true` if the variable exists and is not `NULL`.
6. **What are superglobals in PHP?**
    - Superglobals are built-in variables in PHP that are always accessible, such as `$_GET`, `$_POST`, `$_SESSION`, `$_COOKIE`, etc.
7. **What is the difference between `==` and `===` in PHP?**
    - `==` checks for equality of values, while `===` checks for both value and type equality.
8. **How do you connect to a MySQL database in PHP?**
    - Use the `mysqli_connect()` function: `mysqli_connect('localhost', 'username', 'password', 'database_name');`.

9. **What is the difference between `include` and `require` in PHP?**
    - `include` generates a warning if the file is not found, while `require` produces a fatal error and stops execution if the file is missing.
10. **What is a session in PHP?**
    - A session in PHP is used to store data across multiple pages. Session data is stored on the server.
11. **What is the purpose of `$_POST` and `$_GET` in PHP?**
    - `$_POST` is used to collect form data sent via the HTTP POST method, while `$_GET` is used to collect form data sent via the HTTP GET method.
12. **What is the difference between `unlink()` and `rmdir()` in PHP?**
    - `unlink()` deletes a file, while `rmdir()` deletes an empty directory.
13. **How can you prevent SQL injection in PHP?**
    - Use prepared statements with parameterized queries (`mysqli_prepare()` or `PDO`) to prevent SQL injection attacks.
14. **What is the `$_SESSION` superglobal used for?**
    - The `$_SESSION` superglobal is used to store session variables, allowing data to persist across multiple pages during a user's visit.
15. **What is a cookie in PHP?**
    - A cookie is a small piece of data stored in the user's browser that can hold information such as login credentials.
16. **How do you send an email in PHP?**
    - Use the `mail()` function: `mail('recipient@example.com', 'Subject', 'Message');`.
17. **What is the purpose of the `isset()` function in PHP?**
    - `isset()` checks whether a variable is declared and is not `NULL`.
18. **What is a `foreach` loop in PHP?**

- A `foreach` loop is used to iterate over arrays and objects. Example: `foreach ($array as $value) { echo $value; }`.

19. **What is the difference between `$_SESSION` and `$_COOKIE`?**
    - `$_SESSION` stores data on the server, while `$_COOKIE` stores data on the client-side (in the browser).
20. **How do you define an array in PHP?**
    - Arrays in PHP are defined using the `array()` function or the shorthand syntax: `$array = [1, 2, 3];`.
21. **What is the purpose of `explode()` in PHP?**
    - `explode()` splits a string into an array using a specified delimiter.
22. **What is the `__construct()` function in PHP?**
    - The `__construct()` function is a constructor in PHP that is automatically called when a class object is created.
23. **How do you handle errors in PHP?**
    - Errors in PHP can be handled using `try-catch` blocks for exceptions, and custom error handling can be implemented with `set_error_handler()`.
24. **What is the difference between `echo` and `print_r` in PHP?**
    - `print_r()` is used for printing human-readable information about a variable (usually arrays), while `echo` outputs a string.
25. **What is the purpose of `isset()` and `empty()` in PHP?**
    - `isset()` checks if a variable is set and is not `NULL`, while `empty()` checks if a variable is empty (i.e., `NULL`, `0`, `""`, etc.).

Intermediate PHP Full Stack Interview Questions (26-50)

26. **What is MVC in PHP?**

- MVC (Model-View-Controller) is a design pattern that separates an application into three components: Model (data), View (UI), and Controller (logic).

27. **What is Composer in PHP?**
    - Composer is a dependency management tool for PHP that allows you to manage libraries and packages.

28. **What are namespaces in PHP?**
    - Namespaces allow for logical grouping of classes, functions, and constants to avoid name conflicts in large applications.

29. **What is the difference between `PDO` and `mysqli`?**
    - `PDO` (PHP Data Objects) provides a uniform interface for accessing different database systems, while `mysqli` is specifically for MySQL databases.

30. **What is the `$_FILES` superglobal in PHP?**
    - `$_FILES` is used to handle file uploads in PHP. It stores details about uploaded files, such as name, type, and temporary location.

31. **What is the purpose of the `json_encode()` and `json_decode()` functions in PHP?**
    - `json_encode()` converts a PHP array or object to a JSON string, while `json_decode()` converts a JSON string to a PHP array or object.

32. **What is the difference between GET and POST methods in PHP?**
    - The GET method appends data to the URL, while the POST method sends data in the body of the request, which is more secure and can handle larger amounts of data.

33. **What are traits in PHP?**
    - Traits allow code reuse in multiple classes. They are similar to interfaces but can contain method implementations.

34. **What is the difference between `require_once` and `include_once`?**
    - `require_once` includes a file only once and stops the script if the file is not found. `include_once` does the same, but only generates a warning if the file

is missing.

35. **What is the purpose of the `uniqid()` function in PHP?**
    - `uniqid()` generates a unique identifier based on the current timestamp.

36. **How do you protect your PHP application from Cross-Site Scripting (XSS)?**
    - To protect against XSS, sanitize and escape user input, and use functions like `htmlspecialchars()` when displaying user-generated content.

37. **What is an ORM in PHP?**
    - An ORM (Object-Relational Mapping) allows developers to interact with databases using objects instead of raw SQL queries. Examples include Doctrine and Eloquent.

38. **What is the `foreach` loop in PHP?**
    - A `foreach` loop is used to iterate over arrays or objects. It automatically assigns the key-value pairs to variables.

39. **How do you implement user authentication in PHP?**
    - User authentication can be implemented by using session variables, hashing passwords with `password_hash()`, and validating credentials.

40. **What is the `filter_var()` function in PHP?**
    - `filter_var()` is used to filter a variable, such as validating email addresses or sanitizing input data.

41. **What is the purpose of the `mysqli_fetch_assoc()` function?**
    - `mysqli_fetch_assoc()` fetches a result row as an associative array, where the keys are column names.

42. **What is a prepared statement in PHP?**
    - Prepared statements in PHP are used to prevent SQL injection by separating SQL code from user input.

43. **What is a `closure` in PHP?**

- A closure is an anonymous function that can capture variables from its surrounding scope.

44. **What is the difference between `public`, `protected`, and `private` in PHP?**

    - `public` allows access from anywhere, `protected` restricts access to the class and subclasses, and `private` restricts access to the class only.

45. **What is the `__destruct()` function in PHP?**

    - The `__destruct()` function is a destructor that is automatically called when an object is destroyed.

46. **How do you manage environment configurations in PHP?**

    - Environment configurations can be managed using `.env` files and libraries like `vlucas/phpdotenv`.

47. **What is the difference between `array_map()` and `array_filter()` in PHP?**

    - `array_map()` applies a callback function to each element of an array, while `array_filter()` filters the array based on a callback condition.

48. **What is the difference between `get` and `post` in PHP?**

    - `GET` sends data through the URL, while `POST` sends data in the HTTP body, providing better security for sensitive data.

49. **What is the purpose of the `setcookie()` function in PHP?**

    - The `setcookie()` function is used to send a cookie to the user's browser, which can store data on the client-side.

50. **How can you debug a PHP application?**

    - PHP applications can be debugged using `var_dump()`, `print_r()`, or by using a debugger like Xdebug.

Advanced PHP Full Stack Interview Questions (51-75)

51. **What is Laravel in PHP?**

- Laravel is a PHP web application framework that follows the MVC design pattern and provides tools for routing, authentication, and database management.

52. **What is Composer in PHP?**

    - Composer is a dependency manager for PHP, used to manage libraries and packages within a project.

53. **What are design patterns in PHP?**

    - Design patterns are reusable solutions to common software design problems. Examples include Singleton, Factory, and Observer.

54. **What is the purpose of the `trait` in PHP?**

    - Traits are used to enable code reuse in PHP, allowing multiple classes to share functionality without using inheritance.

55. **How do you create a custom PHP exception?**

    - You can create a custom exception by extending the `Exception` class and defining custom behavior.

56. **What are PHP PDO transactions?**

    - PDO transactions allow grouping multiple database queries into one unit of work, which is either committed or rolled back if an error occurs.

57. **What is the `__autoload()` function in PHP?**

    - `__autoload()` is a magic function that automatically loads class files when they are needed.

58. **What is the purpose of a RESTful API in PHP?**

    - A RESTful API allows communication between different systems over HTTP, where PHP handles the backend logic.

59. **What is Symfony in PHP?**

    - Symfony is a PHP framework used for building complex, large-scale web applications.

60. **What are middleware in Laravel?**

- Middleware in Laravel allows you to filter HTTP requests entering your application.

61. **How do you implement security in a PHP application?**

    - Implement security by validating and sanitizing user input, hashing passwords, using prepared statements, and preventing XSS and CSRF.

62. **What is the difference between `mysqli` and `PDO` in PHP?**

    - `mysqli` is MySQL-specific, while `PDO` provides a database-agnostic interface and supports multiple database systems.

63. **How do you implement file uploads in PHP?**

    - File uploads can be handled using the `$_FILES` superglobal, and the file can be moved to a specific directory using `move_uploaded_file()`.

64. **What is JWT authentication in PHP?**

    - JWT (JSON Web Token) is a method for securely transmitting information between parties as a JSON object, commonly used for stateless authentication.

65. **How can you optimize a PHP application for performance?**

    - Performance can be improved by caching results, optimizing SQL queries, using efficient data structures, and minimizing unnecessary computations.

66. **What are the advantages of using a framework like Laravel over vanilla PHP?**

    - Frameworks provide features like routing, ORM, authentication, and session management, which save time and improve maintainability.

67. **What is the purpose of `xdebug` in PHP?**

    - `xdebug` is a PHP extension used for debugging, profiling, and code coverage analysis.

68. **How does the `MVC` architecture work in PHP?**

    - In `MVC`, the Model handles data, the View is responsible for presentation, and the Controller processes user input and updates the Model and View.

69. **What are the different HTTP request methods in PHP?**

- The HTTP request methods are GET, POST, PUT, DELETE, PATCH, and HEAD.

70. **What is the difference between `$_GET` and `$_POST` methods in PHP?**

    - `$_GET` appends data to the URL, while `$_POST` sends data in the body of the HTTP request.

71. **How do you implement pagination in PHP?**

    - Pagination can be implemented by limiting the number of records displayed per page and calculating the total number of pages based on the total results.

72. **What is `hash()` in PHP?**

    - `hash()` generates a hash value of a given string using a specified hashing algorithm (e.g., SHA256, MD5).

73. **What is a closure in PHP?**

    - A closure is an anonymous function that can capture variables from its surrounding scope and be used as a callback.

74. **How do you handle cross-site request forgery (CSRF) in PHP?**

    - CSRF attacks can be mitigated by using unique tokens in forms and validating them before processing requests.

75. **How do you handle file uploads in PHP?**

    - PHP handles file uploads using the `$_FILES` superglobal, and the file can be moved to a specified location using `move_uploaded_file()`.

Technical PHP Full Stack Interview Questions (76-100)

76. **How can you optimize SQL queries in PHP?**

    - SQL queries can be optimized using indexing, avoiding unnecessary SELECTs, and using prepared statements.

77. **What is the purpose of `ob_start()` in PHP?**

- `ob_start()` turns on output buffering, allowing you to manipulate the content before sending it to the browser.

78. **What are the different types of join operations in SQL?**

    - The different types of joins are INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN.

79. **How do you handle exceptions in PHP?**

    - Exceptions in PHP can be handled using `try-catch` blocks, allowing you to manage errors more effectively.

80. **What is Dependency Injection in PHP?**

    - Dependency Injection is a design pattern where an object's dependencies are injected into it rather than the object creating them.

81. **What is the difference between `include` and `require` in PHP?**

    - `include` generates a warning if the file is missing, while `require` generates a fatal error and stops script execution.

82. **How do you secure sensitive data in PHP applications?**

    - Sensitive data can be secured by using encryption, hashing passwords, and implementing HTTPS.

83. **How does the `session_start()` function work in PHP?**

    - `session_start()` initiates a session or resumes an existing session, allowing you to store and retrieve session data.

84. **What is the purpose of the `mysqli_fetch_assoc()` function?**

    - `mysqli_fetch_assoc()` fetches a result row as an associative array.

85. **How do you handle database migrations in PHP?**

    - Database migrations can be managed using tools like Phinx or Laravel's migration system to handle changes in the database schema.

86. **What is the purpose of an ORM (Object-Relational Mapping) in PHP?**

- ORM allows developers to interact with the database using objects, simplifying database queries and reducing the need for raw SQL.

87. **How do you handle concurrency in PHP?**

    - Concurrency in PHP can be handled by using mutexes, semaphores, or worker queues.

88. **What is the difference between `__get` and `__set` in PHP?**

    - `__get` is called when an inaccessible property is read, while `__set` is called when an inaccessible property is written.

89. **What are the different types of errors in PHP?**

    - PHP has several error types, including Parse errors, Fatal errors, Warning errors, Notice errors, and Deprecated errors.

90. **How do you implement logging in PHP?**

    - Logging can be implemented using built-in PHP functions like `error_log()` or external libraries like Monolog.

91. **What is a WebSocket, and how do you implement it in PHP?**

    - WebSocket provides full-duplex communication over a single TCP connection, and it can be implemented in PHP using libraries like Ratchet.

92. **What is an AJAX request, and how do you implement it in PHP?**

    - AJAX allows you to send and receive data asynchronously from the server without reloading the page. It can be implemented using JavaScript and PHP.

93. **How do you optimize a PHP script for performance?**

    - Performance can be improved by using caching, optimizing SQL queries, using efficient algorithms, and minimizing network requests.

94. **What is `mysqli_real_escape_string()` in PHP?**

    - `mysqli_real_escape_string()` escapes special characters in a string for use in an SQL query, preventing SQL injection attacks.

95. **How do you implement RESTful APIs in PHP?**

- RESTful APIs can be implemented using PHP by routing HTTP requests, processing them with controllers, and

returning JSON data.

96. **What is the role of the `$_SERVER` superglobal in PHP?**
    - `$_SERVER` provides information about headers, paths, and script locations, among other details about the server environment.
97. **How do you handle form validation in PHP?**
    - Form validation can be done using `$_POST` or `$_GET`, along with `filter_var()` to validate and sanitize inputs.
98. **What is a `psr-4` autoloader in PHP?**
    - `psr-4` is an autoloading standard that allows you to load classes based on their namespace.
99. **What is the role of `PHPUnit`?**
    - `PHPUnit` is a testing framework for PHP used for unit testing and test-driven development (TDD).

100. **How do you ensure security in a PHP application?** - Security can be ensured using encryption, input validation, secure authentication, authorization, and protection against common threats like XSS and CSRF.
