PHP – SERVER-SIDE PROGRAMMING

Server-side programming with PHP refers to developing web applications where the PHP code is executed on the web server, before the page is sent to the user's browser. This is in contrast to client-side programming (such as JavaScript), which is executed directly in the user's browser.

We can think of web architecture as a dialogue between two parties:

• Client (User's Browser): The program you use to surf the internet (Chrome, Firefox, Safari, etc.). It requests web pages from the server.
• Server (Web Server): A powerful computer that hosts websites and responds to client requests. It executes the PHP code.
In server-side programming with PHP, the workflow is as follows:

1) Request (Client to Server): The user's browser (client) sends a request to the web server for a specific page. When we type a web address (URL) into the browser or click on a link, the browser (client) sends an HTTP request to the web server associated with that address. The request specifies what the client wants (e.g. a web page, data, etc.).

2) PHP Processing (Server): The web server receives the request. If the request is for a PHP page, the server executes the PHP code. The PHP module interprets and executes the PHP code contained in the file. During execution, PHP can interact with:
• Database: To retrieve or save data (e.g. user information, products, articles).
• Server file system: To read or write files on the server.
• Other server resources: Such as sessions, cookies, mail servers, etc.
3) HTML Generation (Server): The PHP code dynamically generates HTML code (and potentially CSS, JavaScript, etc.) based on the program logic and the processed data.
4) Response (Server): The web server sends the generated HTML page (the result of PHP execution) to the user's browser as a response to the request.
5) Display (Client/Browser): The user's browser receives the HTML and interprets it, displaying the web page to the user.
In summary: PHP runs on the server, generates HTML, and the browser displays the generated HTML. In practice, the Client/Server model allows for separation of responsibilities: the client takes care of the presentation and interaction with the user, while the server manages the logic, data, and resources.

PHP is extremely versatile and widely used for a wide range of server-side web applications, including:

• Dynamic websites: Create web pages whose content changes based on data, user interaction, or other conditions.
• Web applications: Develop complex web applications such as e-commerce, social networks, forums, blogs, content management systems (CMS), etc.
• Database management: Interact with databases (such as MySQL, PostgreSQL, MariaDB, etc.) to store, retrieve, modify, and delete data.
• User Authentication and Authorization: Manage logins, registration, user profiles, permissions, and access.
• Form Processing: Receive and validate data submitted via HTML forms.
• Session and Cookie Management: Maintain user state between requests (e.g. for shopping carts, persistent logins).
• Creating Application Programming Interfaces (APIs): Expose server functionality to other applications or web services.
• Server-Side Task Automation: Run PHP scripts for tasks such as file processing, sending emails, backing up data, etc.
Advantages of Server-Side Programming with PHP:

• Open Source and Free: PHP is free to use and distribute.
• Large Community and Resources: There is a large community of PHP developers and many online resources, documentation, tutorials, frameworks, and libraries.
• Easy to Learn (relatively): PHP has a relatively simple and understandable syntax, making it accessible to beginners.
• Wide Availability of Hosting: Most web hosting providers support PHP.
• Flexible and Versatile: Suitable for projects of different sizes and complexity.
• Powerful Frameworks: There are PHP frameworks (such as Laravel, Symfony, CodeIgniter) that simplify and accelerate the development of robust and scalable web applications.
In conclusion:

Server-side programming with PHP is a fundamental technology for dynamic and interactive web development. It allows you to create powerful and versatile web applications, managing data, users and business logic directly on the

server, before presenting the final result to the user in the

browser. It is an essential skill for anyone who wants to develop modern websites and web applications.