Core PHP vs Frameworks

PHP is by far the most popular server-side programming language for web application development, with nearly 75% of websites using PHP either in its core form or one of the PHP frameworks available. To make a choice between using "core PHP" or frameworks for web development, we need to understand the pros and cons of both.

To give a simple analogy, developing a web application purely with core PHP is like solving a mathematical problem manually by writing down each step on the paper. On the other hand, using a framework is similar to using tools such as a calculator to solve a problem. Just as a calculator, the frameworks are useful tools for rapid application development.

Core PHP vs Frameworks – Pros and Cons

A web framework, especially a PHP framework is a collection of one or more PHP libraries and classes. It provides a generic functionality that allows the developer to concentrate more on the application logic, rather than writing code scratch. It provides a reusable software environment that quickly builds a minimal working template application.

Developing a web application purely in core PHP has its own advantages and disadvantages –

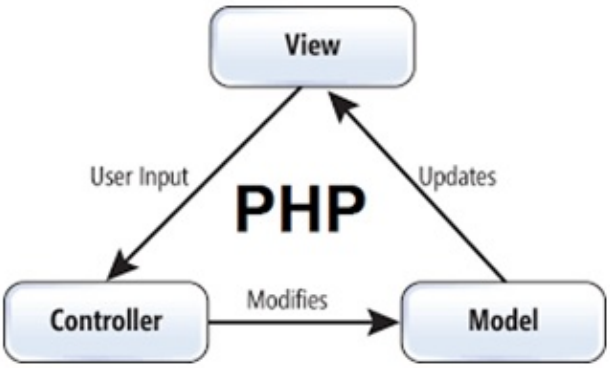
- It gives the developer better control and flexibility.
- At the same time, for a larger application developed with core PHP only can become unwieldy, difficult to manage and maintain.

Now, let's turn to pros and cons of using PHP Frameworks –

- A PHP framework such as Symfony, Laravel or Yii offers a more standardized approach towards web application development. With most of the routine and repetitive part handled by the framework, the developer can concentrate more on the application logic. Hence, there is lesser time wasted in debugging.
- On the other hand, a framework is not as flexible compared to core PHP. The skeleton template of the application is readily made available by the framework, leaving the developer to customize the functionality only within the scope defined by the framework.

The MVC Architecture

Most of the web application frameworks employ the **MVC** (Model, View and Controller) **architecture**, which makes it a lot easier to write quality, robust code, by separating the logic from the style.



If you wish to use the core PHP features for your application development, you are free to adopt an object oriented approach or a modular approach, whichever suits you.

Built-in Security Measures

PHP frameworks offer **built-in security measures** to be incorporated in a web applications.

- If you choose to develop an application with core PHP, you will have to provide the security measures explicitly.
- Most of the frameworks however have a few external dependencies, which may leave the application rather vulnerable, as compared to a core PHP application which is a self-contained solution.

A framework-based application **may be a little slow** when it comes to performance, as compared to core PHP application, especially for a smaller application.

Comparison: Core PHP vs Frameworks

The comparison between the two can be summarized as follows –

- For smaller applications, core PHP is preferrable over framework.
- Framework offers rapid development and code reusability.
- Frameworks are less flexible.
- Using core PHP features, the developer has complete control.
- For large applications, the MVC architecture is helpful.
- Frameworks offer integrated authorization and authentication support. In a core PHP application, the security rules need to be explicitly defined.