Within the scope of my expertise, which isn't much. I choose to utilize the PHP server-side language instead of the others. I am not a strong coder and I think sticking to the main code of teaching for the class would be best for me. Definitely will not roam to others as of yet as I also do not have a preferred coding language.

With vast amounts of server-side languages out there to choose from and some being friendlier than others for beginners to learn, they are dependent upon what you feel the most comfortable with. Server-side languages range from PHP, Ruby, Perl, Lua, Lasso and many more. They all allow for you to write code on a server-side platform such as Apache, Microsoft IIS, Cloud, and MAMP. In regards to server-side development frameworks, I find that they are the workhorses of the code and are the ones that really provide the user with the meat and potatoes of what is truly used by a individual. Some of the most popular ones to note are CakePHP, CodeIgniter, Kohana, Zend, Symfony, Asp.net, Ruby on Rails and the list just keeps going on. It seems like there is a framework for anyone that would feel comfortable utilizing one.

The inherent statelessness of a server-side application refers to the inability for a site to retain users information. Therefore the platform that the programmer utilizes has to use Sessions. This way it will retain the individual user data and maintain continuity of the user information after each logon or session. I have found that this is done by the creation of a cookie or an identifier within the links.

I have found through my reading that some server-side platform pros include things like decreasing the workload from a users machine, dynamically created pages for page creation and that various applications can be built with the utilization of server-side scripting. Though the pros sound good, the cons dictate a separate side regarding the software having to be loaded on the dedicated server, that scripts and CMS need to have databases with the ability to store dynamic data produced from the scripting language and that dynamic scripts can lead to security concerns, therefore making code easier to be hacked.