In the company project we will be using a specific set of tools called a “stack” to develop the application to be tailored in the best way possible to fit our needs. Before we can do that we need to know what stack you feel will suit the project best and to do so we will need to brief you on stacks so you have a better understanding of at least the main ones. The three main stacks our team is considering using are the WISA, LAMP, and MEAN stacks. These three stack names are acronyms that stand for the tools used with them, but don’t get confused, these are just a few popular stacks, a stack can be any combination of tools and may not have an established acronym. We’ll go into details on each of the three stacks and from there you can hopefully make sense of which one you want us to use in development.

First, we have the WESA stack. WISA stands for Windows, Internet Information Server, SQL Server, and ASP.NET. Windows being the operating system we will develop on, Internet Information Server will be our server software used to host the application, SQL Server will be used to maintain a database and ASP.NET will be our backend programming language for writing functions and having an API to connect everything together. One advantage of using a WISA stack is that the application being developed and deployed using Windows, it is extremely user friendly and has little to no learning curve thanks to the popularity of the Windows operating system among the average person. It also has a strong toolset specialized for designing web applications and websites.

Next, we have the LAMP stack. Unlike WISA, the LAMP stack is Linux based and provides a variety of different distributions to choose from. Linux is also UNIX based, and is considered a superior server hosting platform. Apache is the server software and MySQL is the database software. PHP is used as the backend programming language. Because the LAMP stack has such a strong hosting and database toolset, it is used in the vast majority of online websites and web applications. It not only has great server-side potential but it also can allow developers to create some very powerful websites as well. PHP allows for the creation of some dynamic web pages, which have more working parts on a single web page rather than a bunch of simple web pages connected together with links.

Finally, we have my personal recommendation; The MEAN stack. The MEAN stack is far more modern and building momentum quickly in the world of software engineering. The MEAN stack incorporates MongoDB as its database application, Express as a web application framework, Angular as a powerful tool for building dynamic web applications, and Node.js which is basically the front-end programming language javascript, but repurposed for a backend development. MEAN is a powerful, cutting edge stack, largely because you can write the entire thing using Javascript making the development process more streamlined and uniform for the developers. As my mentor said, “javascript is taking over the world” and shows no signs of stopping. The downside to the MEAN stack would no doubt be the learning curve. MEAN is new so there are still many changes being made to the functionality of all of its components and thus you must work a little harder to stay updated with all the changes. Angular is powerful but is known to have a fairly steep learning curve compared to your run-of-the-mill front end kits but for good reason because of how much more can be accomplished once mastered. Node.js also has a tricky learning curve in that it uses Javascript and thus has to deal with the annoyance of asynchronous programming. Asynchronous programming simply means that the code will not read top-down like traditional programming languages do, and will instead process functions In the background while continuing to process the rest of the code in the program. This however has an extremely useful purpose, using asynchronous programming language like Javascript allows the website to run much faster and give it a truly dynamic fluidity that other software just can’t compete with. This is a good reason why Javascript has a nigh monopoly on front end development while also showing signs of dominating in backend development for web applications.

Now that you have an idea of stacks and the choices between them, I trust you will be able to make an informed decision for how you wish for your companies’ product to be developed.