How to Choose a Technology Stack

on July 24, 2019

How to choose a technology stack

You've got a website or an application, it's been humming along for a while, but your company wants or needs to add functionality like video-streaming, e-commerce, perhaps even voice search or automation. Maybe you've even started to notice decreased performance as your traffic grows. Do you worry if your app can handle the increased demands? Do you even know if you have the right technology to implement the new functionality? You need to figure out how to choose a new technology stack that works for you and fast.

You start doing your research on how to choose the right technology for your Web site or application. You read about *coding languages, frameworks*, *front-end vs. back-end*, or come across acronyms and language names that are about as comprehensible as the safety manual for a nuclear reactor.

What do developers mean when they throw out these terms? Let's cut through the confusion and explain it all in plain English.

Stack – This is the unique combination of programming languages and technologies that all stack together to form the engine powering your app or Web site. It encompasses not only the code but your server, back-end software, database, the framework that supports the code, etc. It's the whole package.

Code or Language – Developers write instructions for a computer to take user input or actions and do something with them. They use a special vocabulary commonly called "code" or a "coding language" to write those instructions. Python, Javascript, C#, and Ruby are all coding languages.

Framework – If code is the vocabulary developers use to give instructions to a computer, then a framework might be similar to boilerplate sentences and paragraphs that can be copied and pasted, revised and extended. To put it another way, a framework is simply a method for programmers to reuse and package code for easier and faster development. React is a framework for the popular coding language Javascript.

Front-end – This is the set of coding languages and frameworks that make up what your application users see on their screen. Your front-end stack might include languages like HTML or CSS, or frameworks like Bootstrap. Developers might also refer to *front-end* technologies as *client-side* development.

Back-end – Conversely, the back-end is the mixture of languages, servers, databases, and other technology "behind-the-scenes" that powers Web site processes, sends data, stores information, or connects to other applications. It's also called *server-side* technology. For example, Apache is a common server and MySQL is a database that runs on Apache.

Now that we've gotten a fundamental understanding of a stack and what makes up a stack, let's look at some basic questions you might ask to help you choose a technology stack that's right for your needs.

What type of application or Web site do I have or need?

Your chosen web development company can and should guide you in choosing your technology stack, but they can do that only if you have a good idea of what functionality your app should have.

Are you working with a simple landing page or do you want your site to be the next Facebook? Each has wildly different requirements for a technology stack.

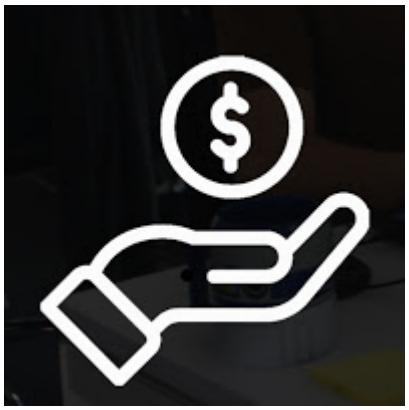
A simple Web site or sales landing page might need nothing more than HTML, CSS, some light Javascript, and *maybe* a small database. You could probably even get away with out-of-the-box content management (CMS) application like Wordpress. Or, if you're doing a minimally viable product (MVP) but plan to build on it later, starting off with Python/Django, might be a good idea given its scalability.

However, if you're interested in large-scale social networking or adding e-commerce, you'll need several languages, frameworks, and server-side technologies to handle your needs.

What is my timeframe for launching my app?

If you're trying to get ahead of the competition and launch quickly, you need a technology stack that can be deployed fast using 3rd party integrations and established, easy-to-use frameworks.

How much do I want to spend on development?



Development costs revolve around three things: development time, developer expertise, and your requirements. If you don't have a lot of cash to spend, you need to choose a team highly proficient in languages and frameworks that can be deployed quickly and easy.

How well-known or well-tested is this technology?

Technology is constantly changing, with the “next big thing” appearing every six months or so. Choosing a technology stack that turns out to be a passing fad could mean wasted time and money if you can’t find developers to add features or make important upgrades using your chosen language. Or, what if you implement a hot new framework that looks great on paper, but can’t handle heavy loads or processing?

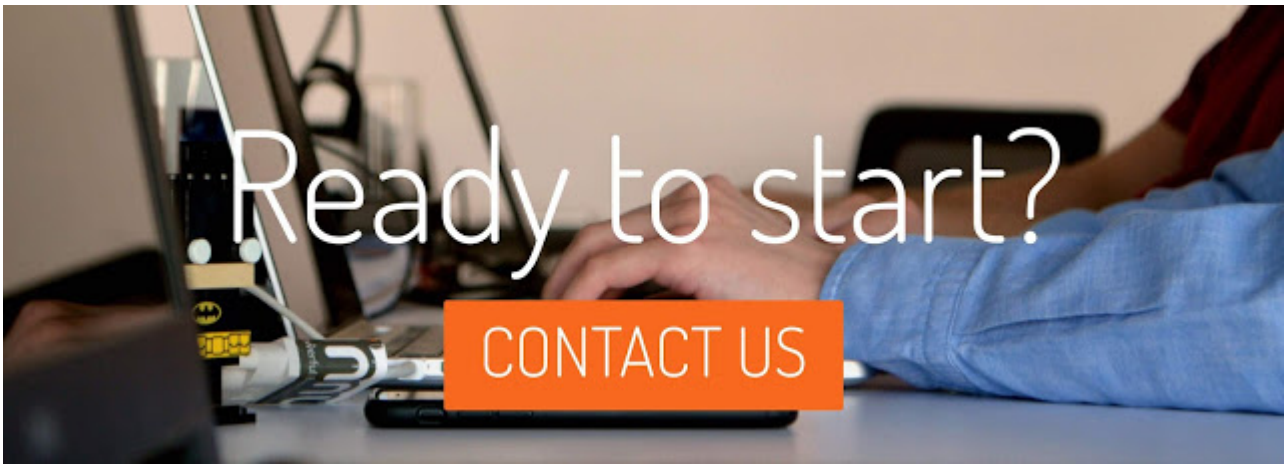
On the other hand, your application might have requirements that only a very niche tech stack can fulfill.

Either way, it’s important to hire a development team with expertise in a wide range of technology stacks to help you choose.

How large is my company expected to grow? How popular do I expect my app to be?

Is demand for your service or app expected to be immediate and intense? What are your plans for growth for your company in a year? Five years? Ten? You definitely want your site to be able to handle demand right away and to grow with you as your business needs increase. Choose a stack that allows you both to scale up (add other software integrations that give you more capabilities) or scale-out (handle more requests and processing).

Here at Innuy, we’re experts in Python/Django stacks and Javascript frameworks React.js and node.js. Not sure if those stacks are right for what you need? Give us a call and we can guide you in choosing a technology stack that works for you.


Ready to start?
CONTACT US