**5 Steps From Front-end to Full-Stack -** [David Minkovski](https://medium.com/@minkovski-d?source=post_page-----3d7d61eda779--------------------------------) Jun 2

A practical guide on how to embrace the back-end

**Motivation**

Have you ever interviewed or talked to someone who has three years of front-end experience and now wishes to go full-stack?  
The market is tough, trust me.  
I have seen many take on different paths from DevOps Kubernetes Certificates to Cloud Serverless Programming, but somehow most don’t end up being Full-Stack Developers. Simply because they focus on popping buzzwords.

That is why I decided to share my personal path for front-end focused developers to become true full-stack developers that can work on full-stack projects within various industries.

**What is a Full-Stack Developer?**

A fullstack developer has a good understanding and expertise in Front-end, Back-end, Databases, DevOps and Mobile Development. Now no one requires you to know everything. You can definitely have your focus. But being fullstack means you understand some technologies in each area and know how everything works together. Software architecture needs to become an area of expertise.

Here is an overview of full-stack technologies:

A screenshot of a computer

Description automatically generated with medium confidence

Designed by the author using figma

**1. Pick a Programming Language & Framework**

Since you are coming from the front-end, I assume you know JavaScript ;)  
And that is perfect! In the past you would not be able to do any server-side coding with that, but luckily there is Node now!  
Now from personal experience I would suggest to learn another programming language for a diversity of thinking and technical problems, but you can definitely build back-ends using JavaScript.  
The framework is far more important. In the end, the language you use will most likely stay the same once you are familiar with it.  
However frameworks can differ very much, even though the core concepts are cimiliar.

A great framework for *JavaScript*Enthusiasts is [Express](https://expressjs.com/). A Node-based web application framework that is quite popular, especially in the startup world.  
If you are open to a new language you should go with *Python*and [Django](https://www.djangoproject.com/).  
And if you like *Java*— the market will thank you if you know [SpringBoot](https://spring.io/projects/spring-boot" \t "_blank).  
You can also go with *PHP*and [Symfony](https://symfony.com/). Some others swear on *Ruby*on [Rails](https://rubyonrails.org/).

Pick one and stick to it! I suggest picking the one you have worked with before, your project is going to need or your friends and colleagues are already familiar with.

***PS****: This will****not****determine your entire future as a developer ;)*

**2. Look for Tutorials on Data Structures & Algorithms**

Data Structures and Algorithms. Personally, this was the most boring course I had to pick at university. But essentially it is important, since this is what differentiates good backend developers from truly great ones.  
In most frontend applications, you don’t need to think that much about efficient algorithms, since you are mostly iterating and manipulating data for the UI / UX.  
But complex calculations are made on the server side. Therefore, the efficiency and effectiveness of your backend algorithms within your programming language decides your level and field of use.

Now again, nobody expects you to know all of them — since most developers copy paste from stackoverflow anyway :P pssssht!  
But it’s important to understand these concepts, to have a good understanding of what problems there are and how they are generally solved.

Some companies issue programming tests within their interviews, so you better prepare by knowing these approaches and practicing on [leetcode](https://leetcode.com/" \t "_blank).

**[Algorithms, Part I](http://Algorithms, Part IThis course covers the essential information that every serious programmer needs to know about algorithms and data…www.coursera.org)**

[This course covers the essential information that every serious programmer needs to know about algorithms and data…](http://Algorithms, Part IThis course covers the essential information that every serious programmer needs to know about algorithms and data…www.coursera.org)

[www.coursera.org](http://Algorithms, Part IThis course covers the essential information that every serious programmer needs to know about algorithms and data…www.coursera.org)

**[Data Structures and Algorithms](http://Data Structures and AlgorithmsMaster Algorithmic Programming Techniques. Advance your Software Engineering or Data Science Career by Learning…www.coursera.org)**

[Master Algorithmic Programming Techniques. Advance your Software Engineering or Data Science Career by Learning…](http://Data Structures and AlgorithmsMaster Algorithmic Programming Techniques. Advance your Software Engineering or Data Science Career by Learning…www.coursera.org)

[www.coursera.org](http://Data Structures and AlgorithmsMaster Algorithmic Programming Techniques. Advance your Software Engineering or Data Science Career by Learning…www.coursera.org)

**3. Learn the Basics of Databases**

Now databases are what store our data. Being in the frontend you might work with cookies, the application store (redux or similar), but it usually ends with graphql and API requests.  
Now the model layer is highly important and differs based on what database technology you are using. Relationships? Documents? Postgresql vs. MongoDB? Different products, approaches on how to structure and query and different benefits and usecases.

It’s important to know those and depending on the framework you choose you will need to know the setup and configurations.

I personally suggest to start with MongoDB.  
The [MongoDB University](https://university.mongodb.com/) has great courses and will help you to start quickly.

[Here](https://www.udemy.com/course/sql-and-postgresql/) is also a great course on Postgresql.

**4. Build Your Own Blog!**

Please, whatever you do…learn by doing!  
Theory in programming is utterly useless without practice.  
You will need to get your hands dirty and start developing everything you learn yourself. I find the best project to learn all these things is your personal website & blog.

This will be fun to you, something you can show and has a long term benefit of maintaining technology.  
You will need a backend framework, a database where you store the posts and maybe more. Furthermore, you will be able to expand the application with more and more functionality. Comments? Tags? This can scale a lot!

**5. Setup Deployment & Hosting**

The final part of going fullstack is to start your own deployment and maybe hosting. Get a server and host your own application on it.  
Setup NGINX, Apache or any web server you like, you can go cloud!  
If you don’t want to setup hosting yourself, you miss out on server configuration and routing problems. This is all part of understanding the fullstack world. One hoster you can use is [A2Hosting](https://www.a2hosting.com/).   
If you still don’t feel like doing that — go for [vercel](https://vercel.com/" \t "_blank)and host for free.

**ToDo List for your Blog**

You need an initial **ToDo List**? Here you go:

1. *Register a domain on*[*GoDaddy.com*](https://www.godaddy.com/)*→ Go for: yourname.com*
2. *Use a front-end*[*blog template*](https://tailwind-nextjs-starter-blog.vercel.app/)*or design your own with [figma](https://figma.com/" \t "_blank)*
3. *Setup your backend and database locally and get the IDE ready*
4. *Create Models for User and Post*
5. *Develop a REST API Controller for the Posts*
6. *Display the Posts in your Blog using your REST API*
7. *Setup a simple UI to manage your Posts and add authentication*
8. *Get yourself a hoster and deploy your application*
9. *Setup DNS on GoDaddy to point to your server*
10. *Publish your website and show it to your friends!*

**Summary**

I hope this article can help you in finding your path to a full-stack role.  
Full-stack developers often have more flexibility to work for different clients because they are able to work in multiple fields. Not only will you be able to apply your knowledge in various fields, but also you will have a better perspective on the front-end and its role in the entire solution.  
Good luck and enjoy the journey!