



THE STANDOUT DEVELOPER

A Complete Guide
for Developers on
Job Hunting, Acing
the Interview, and
Landing the Job

RANDALL KANNA

The Standout Developer

**A complete guide for developers on job hunting, acing
the interview, and landing the job.**

**Copyright © by Randall Kanna
Standout Developer* is a registered trademark**

All rights reserved

The Standout Developer

Randall Kanna

— CHAPTERS —

Chapter 1: Standing Out	4
Chapter 2: Establishing an Online Presence	7
Define Your Niche	7
Leveraging Twitter	8
<i>BUILDING A TWITTER FOLLOWING</i>	9
Junior Engineers	17
LinkedIn	18
Create a Personal Website	22
My TO DO System	25
Chapter 3: Creating a Successful Blog	30
Write a Compelling Blog Post	32
Creating a Content Calendar	38
Promoting Your Content and Finding Your Audience	38
Best Places to Share Your Blog Posts	39
Free Resources for Creating a Great Blog Post	40
Chapter 4: Landing a Speaking Gig	41
Getting Ready	43
How to Get Your First Conference Gig	44
<i>SUBMITTING A PITCH</i>	45
Tips for Becoming a Confident Speaker	46

Chapter 5: The Standout Resume	48
The Fundamentals	49
<i>CRAFTING A POWERFUL SUMMARY SECTION</i>	50
<i>WORK EXPERIENCE</i>	51
<i>SKILLS</i>	54
<i>EDUCATION</i>	54
Editing Your Resume	56
Common Mistakes That Will Get Your Resume Rejected	57
Final Thoughts on Creating a Standout Resume	63
<i>CRAFT A COMPELLING STORY</i>	63
<i>DON'T BE SHY!</i>	63
<i>SHOWCASE YOUR PASSION</i>	64
 Chapter 6: Portfolio	 65
What If You Don't Have a Portfolio?	65
How to Plan Your First Portfolio Project	66
Final Tips to Make Your Portfolio Shine	68
 Chapter 7: The Job Hunt	 71
Tailoring Your Resume to the Job	71
Getting a Job Quickly	73
The Biggest Job-Hunting Myths	74
 Chapter 8: Resources for Landing the Interview	 76
It's About Who You Know	76
Creating Your Pitch	77
The Quick and Easy Guide to Finding a Mentor	79
Meetups	80
CONFERENCES	81
HACKATHONS	82

Chapter 9: The Interview and Beyond	83
The Recruiter Call	83
Passing the Technical Phone Screen	85
The Culture Fit Interview	85
The Onsite Interview	86
<i>GET READY FOR THE “DO YOU HAVE ANY QUESTIONS FOR US?” QUESTION</i>	87
 Chapter 10: Crushing the Whiteboard Interview	 89
Picking the Language	89
Building Your List	90
Your Study Schedule	93
How to Approach a Whiteboarding Problem	95
My Four Hacks to Whiteboarding Interview Prep	98
My Secret for Getting an Offer in Two Weeks	102
 Chapter 11: Cracking the Take-Home Coding Challenge	 104
What Do I Do If I Can’t Finish the Take-Home?	104
Four Things Companies Look for in a Take-Home Challenge	105
Here Are the Best Ways to Debug Quickly	107
A Standout Take-Home Challenge	108
Eight Things That Will Get Your Take-home Rejected	110
 Chapter 12: The Offer	 113
Compensation	113
Negotiating	116
You’re not Married until There’s a Ring	119
 Chapter 13: Conclusion	 120
 Chapter 14: Resources	 121
The Best (and free) Resume Resources	121
Comprehensive List of Job Boards	122

— CHAPTER 1 —

STANDING OUT

How do you become a standout developer?

And why do you need to be a standout? In a perfect world, skill and a good work ethic would be all it would take to land a job as a developer. It's a nice thought, but the world does not work that way. You have to position yourself as a standout.

When I graduated from bootcamp and needed a job, a family friend introduced me to someone at the very top level of senior management at Facebook. This person explained that hiring at Facebook was a “meritocracy,” meaning they only hired people purely for their skill and talent.

That may have been true back then, but today's reality is very different.

Take a look at the developer job listings on AngelList or LinkedIn and you'll see hundreds of developers applying for *one* job. Most if not all of the applicants are skilled and talented.

And for remote jobs you have global competition because applicants can be from *anywhere* in the world.

The job application process is stacked against you: 98% of Fortune 500 companies¹ use applicant tracking systems (ATS) to process job applications, and 75% of those applications² are never read by a human being and are rejected.

You need a roadmap and a proven formula to beat the odds.

It wasn't that long ago that I was one of those applicants applying to myriad jobs and never getting any replies.

¹Over 98% of Fortune 500 Companies Use Applicant Tracking Systems (ATS); Jobscan blog, June 20, 2018.
<https://www.jobscan.co/blog/fortune-500-use-applicant-tracking-systems/>

²Kerri Anne Renzulli, “75% of resumes are never read by a human—here's how to make sure your resume beats the bots,” make it, March 14, 2019.
<https://www.cnn.com/2019/02/28/resume-how-yours-can-beat-the-applicant-tracking-system.html>

Even after I became a developer, my career was held back: It was a common belief that *real* engineers didn't have personal brands and that promoting yourself was something that "good" or "serious" engineers just didn't do. I eventually realized this common belief was misguided.

Writing blog posts, having a website and tweeting seemed like a waste of time and a distraction, and none of it would help my career.

I was wrong.

I developed my coding skills and gained a ton of experience after graduating from coding bootcamp, but my career changed when I became a "standout" developer.

I went from applying to countless jobs and never getting replies to having Google, Facebook, Apple, PayPal, LinkedIn and other notable companies reach out to me constantly. I cannot keep up with the flood of emails.

Imagine having iconic, big name companies relentlessly contacting you about working for them. This book shows you how to get there.

Today I speak at tech conferences all around the world. I lead workshops and have co-written a book for O'Reilly, the biggest engineering publisher in the world.

I'm going to share how you too can become a standout developer, and

- I'll explain how I did it all: what worked, what didn't work and the insights I gained
- I'll tell you how I built my personal brand, credibility, an online presence and a large network
- You'll learn how to be a standout in the interview process and how to create a standout resume and portfolio

- I'll teach you how to get your application **seen** and the strategies I used to get to the top of the pile
- I'll tell you what I did to get on the radar of hiring managers and recruiters at Google, Facebook, Twitter, and LinkedIn and others
- You'll also gain insights from other engineers who will share their unique experiences: Daniel Vassallo, Kyle Shevlin, Marcos Iglesias, Ben Ilegbodu, Abbey Rennemeyer, Shawn Wang and Madison Kanna

Becoming a standout developer will move your career forward, increase your income, and make it easier to quickly land a new job.

Simply put, it will provide you with opportunities.

Just as I was finishing this book, the COVID-19 global pandemic hit. The impact has even been felt in the tech industry, with job listings down 20% according to Glassdoor.³

The things this book teaches are desirable at any time, but given the pandemic and current chaotic job market, I believe standing out is more important now than ever.

Ready to get started?

³J.R. Reed, "Tech companies pull back on hiring, flashing another grim warning sign for the U.S. economy," CNBC, April 23, 2020
<https://www.cnbc.com/2020/04/23/tech-companies-pull-back-on-hiring.html>

— CHAPTER 2 —

ESTABLISHING AN ONLINE PRESENCE

Some of the smartest engineers I've worked with have neither a brand nor an online presence. They are simply coding geniuses. If you bring them a problem, they'll understand it instantly and will quickly have a solution. They just *get* things. Companies love them because they keep their heads down and produce large amounts of work. But they have a hard time landing new jobs or getting promoted.

I worked hard at improving my coding skills, and that was my only focus for a long time. As I shared in Chapter One, I once thought *serious* engineers didn't have personal brands.

But I was wrong. My career didn't take off until I created a brand and established my credibility. The more my online presence and authority has grown, the greater the opportunities that have come my way.

Don't get me wrong—you do need to be a good developer. You need legitimate skills and real talent. But having an online presence and being known in your engineering niche means more companies will discover you and want to hire you.

It's not the coding geniuses hiding within a company who receive the great job offers from marquee tech companies. It's the developers who've built an authoritative brand for themselves.

Where do you start? First, you should establish your engineering niche.

Define Your Niche

Before I realized the importance of defining a niche, I felt I had no career direction. I'd see friends who were also engineers advancing in their career faster than I was, and wonder what I was doing wrong. Then I figured out that they'd established their own identity and niche within engineering.

Shortly after this insight, Blockchain became the hot thing in engineering. Very few engineers were specializing in Blockchain, so I devoted my nights and weekends to learning about smart contracts and Blockchain.

Once I'd established my expertise and my niche in Blockchain, I landed my first speaking gig. Two book deals followed, as well as opportunities to participate in workshops at meetups, at the largest bootcamp in the world, and at company events.

Look at the Twitter bios of engineers who have careers you admire. What about speaker bios at tech conferences? Most likely all or most of them will have differentiated themselves somehow. They have a niche and have tailored their public profile around that specialty.

DO YOU HAVE A NICHE?

Make sure that your resume highlights your experience in that niche.

It's not enough to be skilled in a niche if no one knows you're an expert. Let's look at ways to take that niche and build credibility and visibility.

Leveraging Twitter

Twitter can be an invaluable tool for engineers to advance their career.

I've found countless job opportunities on Twitter and made many new connections. For instance, the last time I was looking for a job, I posted on Twitter asking who was hiring and the tweet went viral. I received countless responses within just a few hours from companies and individuals that were hiring.

I had people I didn't even know offer me referrals and interviews. I was blown away by the response.

Another time I reached out to a meetup organizer and asked if I could speak and pitched them my idea. The organizer immediately accepted my proposal; I was speaking at the meetup a few weeks later. If you want to speak at a conference or you're searching for a new job, conference organizers, hiring managers, and/or recruiters might check your Twitter profile. It certainly wouldn't hurt if your profile showed that you love helping people and that you contribute to the community.

As soon as I had a couple thousand Twitter followers, things started happening in my career. Meetups, conferences, podcasts, etc., and suddenly, far more job offers.

BUILDING A TWITTER FOLLOWING

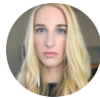
I used Twitter for years with next to no followers. I mostly posted photos of my pets, or memes that I thought were funny. I didn't really think about Twitter as a networking tool and as a result I missed out on some really amazing opportunities. Once I started using a Twitter strategy I landed three podcast interviews, gained over 8,000 followers, and spoke at a conference—all in the same month.

Here's how to implement a Twitter strategy:

1. Provide value in your tweets.

During the time that I wrote this book, I gained 8,000 followers in a month. I did it by making sure that every tweet followed this one rule—provide value. I stopped publishing what I thought was funny or what helped promote my blog posts or my other material, and instead I focused on sharing my own experiences that I thought would add value to others.

This tweet was five years in the making. It was a compilation of all of my favorite CS resources that I had spent hours researching and learning from. The tweet received 9,000 likes overnight and I gained 1,500 Twitter followers.



Randall Kanna
@RandallKanna



I don't have a CS degree so I've had to learn on my own.
Thread on creating your own CS degree online.

5:15 PM · Jun 19, 2020 · [Twitter Web App](#)

 View Tweet activity

5.1K Retweets **18.2K** Likes

Here's a tweet that brought me hundreds of followers, over 7,000 likes, and almost 2,000 retweets.



Randall Kanna
@RandallKanna



Here's a list of resources for developers who want to crush the technical interview.

Thread

8:22 PM · May 20, 2020 · [Twitter Web App](#)



 View Tweet activity

1.9K Retweets **7.1K** Likes

Inside the thread, it had tweets like this, where I shared actual resources that had helped me in the past.

Randall Kanna @RandallKanna · May 20

Replying to @RandallKanna

A very affordable alternate to a CS degree. This list includes incredible book suggestions for learning everything you need to know about CS. I have a ton of these books. teachyourselfcs.com



Tweets that help people learn will be incredibly valuable. I've reviewed a tremendous number of resumes and I shared some of the common issues I've found with them. The tweet below got over 3,000 likes because the lessons applied to everyone.



Randall Kanna
@RandallKanna



I reviewed hundreds of resumes this month. Here are the common issues I found. Thread.

7:32 PM · May 15, 2020 · [Twitter Web App](#)

View Tweet activity

751 Retweets **3.1K** Likes

If you're new to engineering, you can contribute value. When I was deciding what bootcamp I wanted to attend I scoured the web for any information I could find from people who'd attended the different bootcamps I was considering.

I would read their tweets and blog posts and email them about their experience. If you're sharing your journey and you share what you're learning, other people will want to learn along with you and from you.

Don't tweet to make a tweet quota. I decided I wasn't going to tweet unless I felt it was valuable. When I see someone tweeting a lot and I don't feel the tweets are authentic, I stop following them. They seem to be simply trying to reach some quota for the day.

I don't tweet if the content doesn't feel valuable or impactful.

If you have trouble thinking of valuable tweets, try this trick that works for me: I look at my blog posts and figure out ways I can condense them into a tweet while making sure it is still valuable. I determine the key takeaways and share them in a tweet.

2. Find an audience.

It's not enough if you start posting general tweets. You need to start carving out a specific audience that you want to reach. It might be several audiences. I like tweeting about coding, writing, and how people can find developer jobs.

3. Spend time analyzing your tweets that do well.

At the end of each week, I look at the tweets that brought in the most DMs and interactions. Then I brainstorm ideas for more tweets that are similar.

Don't focus on your follower count or what tweets will bring you the most followers. Don't focus solely on having a large audience just to look cool. Quality of followers matters, and engagement with people in your industry and other like-minded individuals will lead to amazing opportunities.

4. At the end of your blog posts, link to your Twitter profile.

I didn't originally do this: instead I linked to my newsletter. I realized that people wanted to get to know me and see what I was doing before subscribing to my email list. Twitter provides the opportunity to connect in a real-time.

5. Build a strong Twitter profile.

You only have a few seconds to catch someone's attention when they click your profile. Make sure you have a Twitter account with a professional handle (preferably your name) and your photo. This will provide credibility and also help you with networking later on. I've been to conferences and had people come up to me because they knew what I looked like. They recognized me from my Twitter photo.

You also need to create a powerful Twitter bio. Your bio is a calling card of sorts. It's one of the first things that people scrolling on Twitter will see. Create a Twitter bio that says you are a software engineer and share what you tweet about, like "React," "Node," or "Security."

Include companies where you've previously worked.

You can also keep your Twitter profile valuable by deleting old tweets and ones that are no longer congruent with your brand, message, or focus.

When someone scrolls my Twitter profile, they'll see a collection of my best tweets, and hopefully they keep reading.

Create a powerful pinned tweet that draws people in. It's hard to post a powerful tweet that draws people in every day. But if you create a great tweet that you can pin to your profile, it can draw new followers in.

6. Answer your DMs.

I spent a lot of time responding to people over direct messages, but I never thought it would impact my Twitter growth. It turned out that the more people that I helped and engaged with, the bigger my following grew. I gained super followers that would retweet everything that I tweeted.

You can also craft tweets based on what people ask you. The best way to add value is to figure out what people want. If a few people start sending you the same question via DM, create a blog post based on that question and answer.

For every person that messages you, there are so many more out there that might have the same question but who don't reach out. This is an easy way to provide powerful help without having to do a ton of research.

7. Give away free stuff.

Giving stuff away has been very effective for me. One of the free things I gave away was resume reviews. When I posted a tweet offering to review resumes, I had no idea the response that it would get. It took a long time and it took up my nights and weekends for a while, but the response from people was absolutely worth it.

Many people didn't know how to put together a strong resume or what hiring managers were looking for in a candidate. It felt great to help them. Consider the time commitment involved in offering a free service that takes up time before you tweet it!

I also wrote a free guide on how to become a software engineer. It covers all a reader needs to know about getting started and what it's like working as an engineer. The guide was downloaded over 1,000 times in one month. This was another free resource I created because I saw a need and wanted to help others getting started in engineering.

Interview with Daniel Vassallo, founder of Uberbase (@dvassallo)

I greatly admire Daniel. He does a fantastic job with his Twitter account. He also has a Twitter course on creating a presence. After I watched his course, using Twitter finally 'clicked' for me. I started looking at Twitter in a completely different way. I used the strategies he outlined and that helped me gain over 8,000 followers in a month.

Q: *You're always sharing high value tweets. How do you come up with so many great ideas?*

Daniel: "In the beginning I dedicated a few minutes at the end of each workday to reflect on what had happened during the day. Whenever I felt I'd encountered something that could be interesting to other people I contemplated tweeting about it. I didn't worry much about staying on one topic: I considered everything that could be interesting. Eventually I started to apply a stricter filter and only tweeted about topics on which I felt I had some credibility. But personal anecdotes remain my favorite tweet type.

This habit of sharing interesting things as they happen became even more natural, and nowadays I tend to tweet about things almost as soon as inspiration strikes. I don't need to spend time reflecting anymore. Many times, I get inspiration when explaining something to someone and I realize that my explanation would be interesting to other people.

Sometimes it's simply when I'm trying to reason about something on my own, and I realize that other people would find my conclusions interesting. Sometimes it's when doing or reading something, and so on. If I could summarize my general approach to Twitter it's to share interesting things that I encounter as they happen."

Q: *What's the one Twitter tip that every engineer should have in their arsenal?*

Daniel: “You're significantly better off as an engineer if other people know about you, and Twitter makes this possible and relatively easy. Obviously Twitter isn't the only way to do that, but I believe it is very likely the medium with the best ROI in terms of effort to exposure.

The attitude to take is to teach what you know, without any expectations. As soon as you learn something that can fit in a 280-character tweet, share it with others. You'll start building your reputation and credibility at scale, and you get to keep it even if you move jobs or start something for yourself.”

Q: *How many more opportunities do you have now because of your impressive Twitter presence?*

Daniel: “Having an audience gives you optionality. I had some great success promoting two info products to my audience—I sold \$180K in 6 months, almost all organically through Twitter—but apart from that I feel that Twitter is an amazing opportunity machine.

I'm getting an average of 140 comments per day from my Twitter audience, which in and of itself tells me a lot about what people want help with, what they're interested in, what they want to learn more about and so forth. I also got to know and meet several interesting people who I would never have had the opportunity to encounter if it wasn't for Twitter.

I had been on Twitter for years before that and only had a few hundred followers. Mostly my friends! But as soon as I started sharing tweets about job hunting and preparing for the coding interview, my tweets went viral.

Start interacting with other engineers on Twitter and responding to their tweets. There's no point in being on Twitter if you aren't creating new connections and forming bonds. If you aren't interacting with other users and getting interactions from them, your tweets are basically going into the void. I've met a ton of friends on Twitter and was able to get some incredible opportunities. Don't just reply with any statement on someone's tweet. Make sure that you're adding value.

If you're a junior engineer you can write tweets about your coding journey. People want to know what worked and what didn't. When I started out in engineering I loved reading blog posts and tweets from junior developers to see what worked for them on their journey to becoming an engineer."

Junior Engineers

If you're a junior developer and looking for a new job, craft a tweet asking for advice and information. Sharing your job search on Twitter can be a very powerful strategy for creating connections and getting retweets. People want to help you.

If you don't open a Twitter account, I believe that it will be much harder for you to leverage the engineering community on Twitter.

JOB-SEEKING TWEETS

Craft out a tweet that is short, simple and to the point. People scroll past tweets quickly, so you want to make sure your tweet isn't too long. If you have a required location or you only want remote, make sure you state that.

Here's a job tweet that I created that went viral.

"I'm currently looking for a frontend or full-stack developer position. 5+ years of experience. Remote only. I've worked at companies such as Eventbrite and Pandora. DMs are open. Please retweet for reach!"

List out what kind of job you're interested in, and what technologies you're experienced in. Is the majority of your experience in backend or frontend?

This will help people who are quickly scrolling through Twitter to know if you're a fit for an opening they might have heard about.

Do you have any impressive facts? Include them. I've worked as a frontend, iOS, full-stack and Blockchain developer, so I made sure to add that to my job search tweet. What is something that makes you unique?

After you've created your job hunting tweet, make sure to pin the tweet to your Twitter profile so it's the first thing people see. Every week or so, craft a new tweet that's worded a little differently. Try different approaches to find out what works and what doesn't.

LinkedIn

LinkedIn is your online resume. Most engineers have a LinkedIn account and it's crucial that you do as well. When I was trying to get my first developer job, the company that ended up hiring me was the one that found me through my LinkedIn profile. My profile ranked high in search keywords.

Here's how to make your profile stand out.

YOUR HEADLINE MATTERS

This is the first thing that recruiters see on search. You want to make sure your headline highlights your most impressive accomplishment and your current job title. My bio includes that I'm a published O'Reilly author, as that's the accomplishment that I most want to share with the world.

If you want a job working in React, write frontend developer or react developer in your headline. Keywords matter on LinkedIn search, and having a relevant headline will help you increase your odds of getting found through search. LinkedIn has a feature that will show you how many times you appear in recruiter searches, and you want to optimize that as much as possible.

If you don't have an engineering job you can state, "Looking for frontend opportunities." If you don't feel comfortable adding that, you can change your headline to "Junior Engineer" so you can show up higher on search as well.

WRITE A COMPELLING SUMMARY

When people check your LinkedIn, they're going to scan your headline and photo and then they'll quickly look at your summary. This will be your opportunity to succinctly express a little about yourself and what you're currently looking for in a position.

If you have a job, you can set "Open to new opportunities" on LinkedIn privately so only recruiters will see it.

USE THE "ACCOMPLISHMENTS" SECTION TO YOUR BENEFIT

For developers without a traditional computer science degree, this is the section that can make or break your profile. This will emphasize your passion to recruiters and make your profile stand out.

When I started out as a junior engineer I made sure to add every course I'd taken and every certification I'd earned to my profile. Completing more to be able to add was a good motivator. Make this section as robust as you can and keep updating it.

While I didn't have a CS degree, I did make the dean's list for multiple years in college so I included that. And while I didn't have any work experience as a developer yet, I had spent the last eight months taking online courses and earning certificates.

Think of all your professional wins and all the extracurricular courses you've taken and add them to your profile.

Do you have a portfolio? You can include this under "Project" in the achievements section. Doing this was a huge help in getting companies to email me back when I was still a junior developer with no job experience.

FILL OUT EVERY SECTION ON LINKEDIN

If you fill out every section on LinkedIn, not only will you be more visible on search, but you'll also reach "All-star" status. All-star users are 40 times more likely to get contacted via LinkedIn, so it's worth the investment.

You'll need to make sure you have the following filled out to gain All-star status on LinkedIn:

- ☐ Profile photo
- ☐ Experience
- ☐ Skills
- ☐ Education
- ☐ Connections - Try to have at least fifty LinkedIn connections
- ☐ Summary
- ☐ Industry and your current location

GET A CUSTOM URL

If you don't have one yet, you're probably using the default LinkedIn URL. You don't want to use that on your resume so it's important to change your URL to your name.

My custom URL is [linkedin.com/in/randallkanna](https://www.linkedin.com/in/randallkanna). A custom URL will help make your resume look more professional.

ASK SOMEONE TO WRITE A RECOMMENDATION FOR YOU

This is a great way to build a little credibility. Recommendations show recruiters that you're a real person and can do the job well.

It's not the end of the world if you don't have a lot of recommendations, and it probably won't matter at all later in your career. But if you're a junior engineer, having recommendations can be extremely helpful in getting interviews and a job offer.

Ask a past coworker to write a recommendation for you. Offer to write one in return as well.

USE A PROFESSIONAL HEADSHOT

A headshot makes your profile much more likely to be viewed on LinkedIn. This is an important part of creating your own personal brand as an engineer.

It's a pain to worry about getting a good headshot taken, but doing so will dramatically increase your chances of getting called in for a job interview.

Don't use a selfie or the same photo you use for Tinder. Ask a friend with some photography skills to take your headshot and Photoshop the background out if needed. Make sure the photo is close up, use a simple background, wear professional clothes, and make sure the lighting is good.

Some meetups even offer days where you can get a professional headshot taken for free. Certain companies and stores also offer days where you can visit and get a free professional headshot.

Create a Personal Website

A personal site is one of the best things you can do to build your engineering brand and one of the most impactful tools for creating a standout career.

A personal site, if done well, can create credibility and showcase your work.

It can also help sell you to employers and help enrich the dev community. There are a ton of benefits to having a personal site, and it doesn't take that long to get one set up.

Unless you have a very unique name, when you're applying to a job it's hard for an employer to find much information about you online. But if you have a personal site, you can link to it on your resume and showcase your best projects and blog posts to recruiters and companies.

Setting up a website can be a simple, straightforward process. I've laid out the steps so you can get closer to building your brand.

BUY YOUR DOMAIN

The first thing you need to do is buy your domain name. I love using Google Domains because you can easily set up an email address and Gmail account with the domain you purchase.

My domain is RandallKanna.com, which isn't a common name and my Mom bought it for me a long time ago. Having your name as your domain is ideal but it may not be available, so be creative but professional. I love seeing creative domain names such as BuiltBy\${yourName}.com or \${yourNickName}.dev.

BUILD YOUR SITE

After you've purchased your own unique domain name, it's time to build your personal site.

If you don't want to spend a ton of time creating a personal site yourself, you can use WordPress or Webflow. Webflow is really cool because you can pick an existing template and easily customize it. You can even create your own template from scratch using Webflow.

If you're up for coding your own personal site from scratch, that's awesome. I love using React to quickly create a personal site.

You don't have to create your personal site from scratch if you don't want to invest the time. In fact, my latest site is built on WordPress for convenience. I also love using Webflow when I want to deploy a site quickly. But it's important to have a few key pages on your personal site.

Make sure your personal site includes the following:

Your personal site should include a blog. A blog is your way to start building your engineering brand and voice. (I'll cover creating your first blog post and more in the next chapter.) Unless you have a very unique name and you're easy to find on Google, it's important that you include your blog posts on your personal site so they're easy for recruiters to find. You can then link to your website on your resume.

1. Blog

Your personal site should include a blog. A blog is your way to start building your engineering brand and voice. (I'll cover creating your first blog post and more in the next chapter.) Unless you have a very unique name and you're easy to find on Google, it's important that you include your blog posts on your personal site so they're easy for recruiters to find. You can then link to your website on your resume.

2. Contact Section

When companies find you through your blog post or Twitter account, you'll need to have a way for them to easily contact you to discuss job offers. You can send them to your LinkedIn or include an email address on your website, or even include a contact form.

3. Projects

Your personal site should house all of your portfolio projects and include a summary for each one so recruiters and companies can see what you've created. Recruiters shouldn't have to search through your GitHub profile to see your best projects.

4. Relevant links

Here are a few examples of links to include:

- GitHub profile
- Any open source repositories you contribute to
- LinkedIn
- Twitter

GET FEEDBACK BEFORE YOU GO LIVE

Before your site goes live, ask a friend (ideally a friend with design experience) to make some suggestions.

Design is not one of my strengths. Whenever I create something for a side project I ask (beg) one of my designer friends to take a look and make some suggestions. If you have the resources, hire a good designer. Building an online presence requires good visuals. A good designer can be priceless.

DEPLOY!

You can easily deploy a personal site using Google Firebase. I've created a site on React and was able to quickly add my domain to Firebase and deploy my React site. There are a ton of tutorials out there and options for deploying your own site.

If you selected Webflow, it's easy to add your custom domain to your Webflow project and deploy it from there. There are many [tutorials](#) online that will help you use a custom domain with Webflow.

An essential part of building an online voice and presence is creating an authoritative blog. Chapter 3 tells you how.

Before we move on to Chapter 3, let's talk about ...

Finding the time, systems, the discipline required to do what it takes to be a standout engineer.

My TO DO System

When I first started out in engineering, I thought a senior developer promotion was years, if not a decade, away. So, I didn't set it as a goal because it seemed so far in the future.

Then, about a year into my engineering career, I realized I was already doing the work of a senior engineer; I was helping lead projects, overseeing giant pieces of applications on my own, reworking hiring processes, and helping organize teams. I was contributing as much code as a senior engineer.

So I asked for a promotion. I presented my manager with a comprehensive document on all of my achievements and work at the company to date.

I got the promotion.

Here's the process I created to manage my time, stay disciplined, and get more done to get that promotion. I still do this every day.

My standup updates and productivity changed drastically as soon as I used a note-taking system that helped me to stay organized and focused.

Here's an example of my daily to do document in Evernote.

Thursday, July 18, 2019

Standup

A quick summary of what I did the day before so I can report quickly to my team and also have a record of this later for interviewing.

Meetings

- ☐ Standup
- ☐ Book club

3 Big Daily's

- ☐ 2 hours of book writing. Finish both chapters this week.
- ☐ Start my next big feature
- ☐ Get all my current PR's finished and merged

Daily work

- ☐ Review pull requests
- ☐ Review my notes for tomorrow's 1-1

To Do

- ☐ Walk the dog
- ☐ Eye app

Habits

<input type="checkbox"/> Gratefuls	<input type="checkbox"/> 2 hours of writing
<input type="checkbox"/> Drink 2x water bottle	<input type="checkbox"/> Print daily ToDo's for tomorrow
<input type="checkbox"/> 30 minutes of reading	<input type="checkbox"/> Focused work first
<input type="checkbox"/> Tidy house for 15 minutes	<input type="checkbox"/> Bed by 11:30PM
<input type="checkbox"/> Brush the dogs teeth	<input type="checkbox"/> Meditate

Standup. At the end of every day, before I shut off my computer for the night, I spend a few minutes writing a summary of all the work I completed that day. I write down each ticket I completed and my big wins of the day. Doing this at standup the next day helps me to not struggle to remember all the things I did, and it helps me to be better prepared for future interviews. You can look through your old notes and see what you worked on and achieved. It is much easier to add these highlights to your resume when you can review all the wins you've documented.

Meetings. Like many many people, I hate meetings. So, I like to check them off one by one throughout the day to gain a sense of accomplishment after having attended them. I take notes on this section if I need to remember something for a particular meeting.

Three Big Daily's. (My most important tasks.) I define the three big tasks that I need to get done each day that will help me move forward in my career and life. These are the important tasks—the ones that will make you feel like you've been productive, even if you get nothing else done for the rest of the day.

Daily Work. This is the mandatory task that you need to do every day for your job.

To Do. My personal task list with all the things I need to check off that day.

Habits. I try hard to improve my habits daily, so keeping a list of them to check off after I complete them keeps my habit streaks on track.

TAKING BREAKS

When you get stuck on a bug, the worst thing you can do is endlessly stare at your screen and force a solution. I've spent hours trying to fix a bug and stayed completely stuck. I'd take a break, go for a walk, or eat something and take my mind off the bug. After returning to my computer I was able to fix it immediately.

I use the Pomodoro technique to stay focused and to remember to take breaks.

When I don't follow the technique, I find myself taking too many breaks and wasting time by aimlessly scrolling Twitter. Or, I don't leave my computer for several hours and my productivity plummets because I haven't taken a break.

Set a timer for 25–45 minutes and stay focused on **one** task at hand. No distractions, do not check social media, texts, etc. After the timer rings, take a break for 5–15 minutes. Start another pomodoro session after your break. I find that I can do about 4–6 pomodoro sessions a day.

HOW I JUST ‘KNEW’ EVERYTHING AFTER BEING TOLD ONCE

At my first engineering job, instead of needing a senior engineer to show me something several times, I started taking detailed notes.

I would copy and paste the exact terminal commands if they showed me how to deploy something, and I would take notes on everything I learned that day.

This process provided me with a comprehensive document of everything I needed to know to quickly solve issues. Instead of needing to ask someone again what a command was or how I could quickly fix an issue with the deployments, I was able to simply search through my own document.

In this document, I include a header with keywords to quickly search through my notes. Sometimes I write: “Remember that you need to change to the QA environment instead of production when running that command.” Doing this came in handy many times and avoided creating a production issue.

I’ve actually found this habit so useful that I’ve kept it up throughout my career. To this day, I keep an Evernote folder of everything I need to know about my current company.

DON’T FAKE IT UNTIL YOU MAKE IT. DO THIS INSTEAD.

“Fake it until you make it” is a common saying, and you hear it often in engineering. Pretend to be confident. Pretend you always know the answer.

But here is the problem. You'll never be able to ask questions and expose what you don't know. Faking it until you make it might work for a short while but at some point, people will realize that you don't know everything.

And that's okay.

It's better to realize sooner than later that no one knows it all. Even the most senior engineers and tech leads ask questions. No matter the level of experience, everyone has gaps in their knowledge.