

Ask HN: Advice for finding an entry-level remote job?

740 points by AskHNremote2021 17 days ago | [hide](#) | [past](#) | [favorite](#) | [199 comments](#)

Hi HN,

I realize this isn't super on topic but I also feel like this is the best place I know of to ask for this advice, so here goes. I need some entry-level, remote-based work. What should I do? Help desk work seems the most promising / practical, but I haven't been able to find anything yet. The remote jobs I see posted are nearly all for higher-end positions.

I live in the poorest region of the United States, but I do the best with what I have. I've worked on my family's farm and done a couple stints at retail beauty supply shops that friends own. I helped open two of those shops. That's the extent of my non-existent resume. Given a chance to interview, I believe I would do ok. Maybe even exceed expectations for the sort of job I'm looking for.

I need to work remotely for family reasons. I have a special-needs sister and I look after my youngest brother. They are what's most important to me, which is why I don't want to relocate. I have another brother who was helping me, but he accepted a job offer far away. Now I am the only relative near who'll be able to care and look over them. I have time for a full-time job, though, and I need a way to support us.

I do have a job offer that would require me to move by March 13th. The problem is that it is far from my family, and with my brother gone, I would be leaving them on their own. The job is at an auto body repair shop paying minimum wage. I would be stressed every day worrying about my family back home. What I want is a way to work that lets me stay at home, fulfill my family responsibilities, and make money to keep things afloat.

I am a techie at heart. I'm a Linux/macOS person, but I easily adapt to other technologies. My first PC was a Compaq Presario that ran Windows 3.1. My father saw the 'future' in it, and he hoped I would be part of that future. To use it, you needed to enjoy torture to some extent. Still, it sucked me in. Something about that mysterious DOS prompt promised treasures if only I learned its magic. A few years later, I was dual-booting an ugly Dell machine (Windows 98 SE and Ubuntu). In between that time, my school still had an Apple IIe on which I loved playing Oregon Trail. I bought one a few years back for nostalgic reasons, but I had to leave it behind at my old residence. I miss it a lot. I hope it got a good home.

I am currently working through the freeCodeCamp course and intend to pick up Eloquent JavaScript by Marijn Haverbeke soon. I use VS Code and Spacemacs as my editors. I google like a madman. I have fun playing ukulele and guitar, and I'm teaching my youngest brother about the different parts of a Raspberry Pi. Oh, and I love to read. I am a habitual reader. There's a lot more, but those are the kind of things that interest me.

I learn quickly, I am flexible, and due to working in customer service (beauty supply shops) I have a calm and understanding demeanor. I am a friendly person and I am always willing to find a solution, even when a solution seems impossible.

I would be grateful for any advice, and I am particularly thankful for dang's / Daniel's time in editing this to be a better Ask HN submission.

I can be contacted at AskHNremote2021@gmail.com and I can provide my GitHub as well, which is mostly documentation editing. I have been told I am a competent writer, if that counts for anything.

Edit: I know that this is an unusual Ask HN post and I am grateful to anyone who takes the time to read through it. I'm curious how others in my situation managed to find remote work. I feel lost in all of this. To say that this has been a stressful time would be an understatement, but I'm turning to HN in the small chance the right person sees this and can give me suitable advice or point me in the right direction. I have always found comfort in this community, so this is where I've turned. Thank you.

geocrasher 17 days ago [-]

I am sorry about the position you're in. For the last several years I have been balancing taking care of my terminally ill wife who recently passed away, all while working remotely in the web hosting industry. It's hard to balance. Harder than you think. Setting boundaries will be vital to being successful in any WFH role.

Now that that's out of the way: You need to focus. You've described your abilities and that's fine, but you lack focus. Your qualifications are great, but your presentation is all over the place. You need a single one page resume that focuses on *one* aspect of your knowledge and experience, and lightly mentions the others. Make multiple resumes, each with a focus on one thing. Use the resume that's best for the job. I recently saw somebody who must have had 20 different resumes to match the jobs they were applying for. It worked.

Also, you might consider looking into the Web Hosting business. If your temperament is as you describe, and you like talking on the phone or live chat, most web hosts will overlook any technical gaps. Those can be taught. In fact that can lend to your focus! Focus on the fact that you know things that *can't* be taught: Customer Service, talking down angry customers, talented writing, friendly and always willing to find a solution under dire circumstances. Those are GOLD. Once those are on the table, the rest is negotiable. I'll email you a link to at least one that is hiring full remote entry level.

Additionally, you need to step up your confidence level. Let an employer know that your family is important to you and that you're dedicated to working hard so that you can support them- and leave it at that. They don't deserve the other details, and they aren't relevant.

I hope this helps.

fecak 16 days ago [-]

Resume writer here. Don't overdo the resume customization thing - I've had clients who wanted to spend lots of time tailoring each line of a resume to the job, and the time invested really isn't worth it. Especially if it becomes far too clear to the reader that you were essentially pandering.

For most people, different resumes for different roles (web dev, mobile dev, etc.) make sense, but if you have 20 resumes for different roles you're probably applying for jobs that you are highly unqualified for - nobody has expertise in so many areas where they need 20 resumes. 2-3 for most is probably plenty.

The key difference in most resume customizations is how you define yourself to the reader. I use summaries to tell the reader who I WANT them to think you are. If the job is "web developer", I want to tell the reader right away that you are a web developer right in the summary. You've defined yourself as what they are looking for, checked their first (and most important box), and they will now keep reading.

If that summary said "embedded developer", they may stop reading. If there was no summary and your first job title is "data analyst", they may stop reading.

devoutsalsa 16 days ago [-]

I'd also add that someone hiring a green person for entry level work may be looking for enthusiasm, aptitude, and general excitement. Stripping all that away may be detrimental. There was definitely a time where I had no real experience to speak, and plenty of times I feel like my experience isn't worth the time I spent writing a resume, and if I'd stripped out the passion I had, I'd might still be doing crap work. If you have little experience to focus your focus, some enthusiasm sounds great to me!

AlexDanger 17 days ago [-]

>Make multiple resumes, each with a focus on one thing. Use the resume that's best for the job. I recently saw somebody who must have had 20 different resumes to match the jobs they were applying for. It worked.

I'd like to second this methodology. It is a very effective approach. I've worked across multiple specialties within technology and I always tailor my CV to the job I am applying for. I highlight the most relevant experience and skills required and remove (or summarize) experience that is not relevant to the position.

Matching your CV to the job can be as simple as updating it to use the same keywords that you see on the job advert. It sounds trivial but many recruiters will filter job applications with a keyword search. If they don't see the exact keywords they are expecting, your CV will be thrown away.

Good luck!

mettamage 16 days ago [-]

Question: how do you deal with gaps in your resume?

Let's suppose you were a machine learning engineer for 2 years, a web developer for another 2 and a product owner for 3 years.

I presume that some experiences would almost make it seem that you're not dedicated to the role you're applying for if you leave them in. So you have to leave them out.

How do you deal with that?

ldjb 16 days ago [-]

That's an interesting question and something that hadn't really occurred to me.

Personally, over the past few years I've been a developer, a product owner and now a product manager.

But if I were to apply for a new developer role, would I leave out my current product manager role from my CV? Absolutely not!

It's generally considered a bad idea to have gaps in your CV. And actually having worked various roles can actually be to your advantage. There are transferable skills you would have gained as a product owner that you can apply to a developer role. In fact, very few developers will have the skills required to be a product owner, so employers should look favourably upon your skillset.

Case in point: my experience as a developer has put me at quite an advantage in the product management field. Due to my technical knowledge/experience, I can understand how systems work better than most other product managers, and I'm better placed to know what the development team needs of me.

So I would say, don't leave gaps in your CV, include all the really good experience you have, even if it is varied, and use that to your advantage to show how these experiences will help you in the role you're applying for.

klondike_klive 16 days ago [-]

How about if I was a stay-at-home dad for a year? I haven't included that in my resume - I figured if I were asked I have no problem speaking about it, I'm actually super proud of it, it just doesn't strike me as something I should put in my list of experience.

killtimeatwork 16 days ago [-]

Employers worry about gaps in CV because of two possible scenarios:

- you were actively looking, but were unable to find a job during that whole time - for them it's social proof that you suck
- you're actually picky and "high maintenance"

DuckBoy 16 days ago [-]

"you were actively looking, but were unable to find a job during that whole time - for them it's social proof that you suck"

Gee... I would hope I don't suck. Most places I apply to ghost me when they find out I'm deaf! They don't want to "deal with accommodations".

ldjb 16 days ago [-]

I think that's a perfectly reasonable reason for you to have a gap in your career, and prospective employers should be understanding of that.

It is something you should be prepared to explain at an interview, but I don't think it would cause any problem.

You're right that it doesn't really count as experience. Employers won't necessarily mind a one year gap on your CV, but if you do find it difficult to get to the interview stage, it could help to add a short note to your CV explaining the gap. But I'd hope that wouldn't be necessary.

blackrock 16 days ago [-]

How's the compensation (pay and/or stock options) for your product manager role now, as compared to your development roles of past?

fecak 16 days ago [-]

What you are referring to aren't what we'd traditionally call gaps. Resume gaps are usually periods of unemployment. What you're referring to seems to be some career inconsistency.

I write resumes all the time for people who have bounced between different types of roles. In those cases, having a couple resumes makes sense. For the description you have in your comment, if this person is applying for ML roles, we provide more detail and emphasize that role. If they were applying for a product owner job, you emphasize that aspect and those skills.

As I mentioned in another comment, you can also use a summary to define yourself. If you're applying for product owner jobs, you define yourself as a product owner with skills in web dev and ML. If you're applying for a web dev job, you define yourself as a web dev with skills in ML and product.

redshirtrob 16 days ago [-]

Be honest. I make a habit of having gaps in my resume. I like to take some time off between jobs. It's a luxury, but if you can swing it, it will do a lot to recharge you and get you ready for your next role.

If the gap is fairly large (say over six months), be prepared to discuss what you were up to during that time. Hopefully it's something interesting, but it doesn't have to be something related to your career path.

To your actual question: I see no issue with bouncing between roles. If anything it shows flexibility and a diverse interest. You'll fit right in at small to medium places where folks frequently wear many hats. Plat it as a strength.

T-zex 16 days ago [-]

Why would you make such an absurd presumption? Life is life and there are lots of circumstances pushing you to make tough choices. Always being able to stick to your dream job or career path is not an option for most people.

mettamage 16 days ago [-]

It's my experience when applying for a job that people are very judgmental. I've learned to take that judgmental frame, if I wouldn't then I would never be invited for an interview.

The only places that don't have these judgmental frames are a subset of startups.

Just my experience, YMMV.

AlexDanger 16 days ago [-]

fecak's reply is the best answer. In particular this quote:

>As I mentioned in another comment, you can also use a summary to define yourself. If you're applying for product owner jobs, you define yourself as a product owner with skills in web dev and ML. If you're applying for a web dev job, you define yourself as a web dev with skills in ML and product.

This is great advice. Make sure you still include all the roles in your CV to illustrate a consistent period of employment. Just highlight the aspects of each role that are most relevant to the position you are applying for. If a previous role is barely relevant, just summarize the experience in a couple of sentences.

The point is to make most of your CV relevant to the role. Dont make the employer hunt through your CV for relevant experience. Most people will make a snap judgement within a minute or two of scanning your CV. Make it easy for them.

I also think a summary is very important. Sum up your best points with some dot points at the top of your CV.

de_nied 16 days ago [-]

This seems extremely helpful. 20 resumes, as GP mentioned, should be an excellent source. I've always seen multiple cover letters, and maybe 2-3 resumes, but never 20. While 20 is fairly high, I think it's also the right amount in being able to have a resume for almost every position that is "just right."

audiometry 17 days ago [-]

It's pretty kind of you to take such effort to write this fellow a detailed reply. Especially considering the crappy situation you've had to deal with. Hope 2021 is a better year for you.

mattbee 17 days ago [-]

Hey anon, I feel for you, and most of the answers here are giving you a 12-month plan when you need a 1-month plan.

If you know Linux well enough to configure Wordpress, how about tech support for a hosting company? Try making some short email approaches to old-school hosting companies (I used to run one). They might be flexible enough to take a chance depending on how confident your approach is.

They have lots of customers bashing away at Linux, making mistakes, often without the patience to see their own problems through. Their business problem is that these customers need hand-holding but only pay a fixed, monthly fee. (the hope is eventually they stop asking and keep paying for years).

The combination you can offer those companies could be basic Linux knowledge (no need for advanced cloud stuff) and whatever flexibility you can offer them - especially if they're not in your time zone.

The larger ones might be a tall order, all listing locations by default (Gandi, Leaseweb, GoDaddy, Hetzner etc.), but maybe someone here will have an inside track.

I agree with other posters - never mention difficult circumstances in a job application, particularly a cold application. Just talk about how keen you are to solve their difficult customers' problems, how well you work in a team, and keep the initial approach brief.

Good luck! And please update us if you can.

whoknew1122 17 days ago [-]

This is the advice I received when I first got into tech. Look for managed hosting services that run off Linux. WordPress, Minecraft, or just about anything that people sell as a managed service that runs off Linux. Another place to look into is web registrars.

Once you get that first job, just soak up as much as you can. And keep studying on your offtime. Learning AWS is very valuable.

brandall10 17 days ago [-]

To add to your last comment, there are relatively cheap courses to help get AWS certifications on places like Udemy.

wyclif 17 days ago [-]

The best gambit is to get an entry-level job that will pay for your AWS certificates while you work. Always, always, take advantage of free training when it's offered.

brandall10 16 days ago [-]

We're talking like \$20. Considering the situation the OP is currently in, I'd consider that money well spent to aid in getting that first job and assisting in leapfrogging to the much better one soon after.

wyclif 16 days ago [-]

OK, sure. Granted, it's not expensive. But at least if a company is willing to support you in getting the AWS certificates, it means they're willing to help you grow as a developer.

brandall10 16 days ago [-]

I get your point in a general sense, but if I were in the OPs shoes I would certainly not wait for or expect something like that. He's in crisis mode. He's looking for any job that will allow him to remain at home with his family. He will have to hustle to get that first job which likely won't have any interest in supporting growth like that. That's why I said "leapfrog to the next better job".

Back in the day instruction/certification for things like this would run in the thousands/be in person would be a non-starter in a situation like this. We're in a different world now where instruction like this has an infinitesimally small cost compared to the value it can bring.

The question is whether or not doing this now is the best use of his time, and I would argue yes, absolutely, the earlier the better. Just the education alone is powerful, but having an actual AWS Certified Developer cert will open doors.

collectorof 16 days ago [-]

Somewhat off-topic, but important.

Note how this path wouldn't be possible if all these hosting companies went out of business thanks to competition from big cloud providers. Then think about what this means for the future we (in IT) create for ourselves and for newly minted IT professionals.

I commonly see people on HN deny that there are any major shifts in overall system complexity happening right now. "Everything always seems complex if it wasn't created when you started in IT". Ok. If the complexity of newer tech stacks is an illusion, why don't you hire the OP to manage your containerized micro-services via Kubernetes? He's starting his career in IT right now, so he should have no preconceptions you allude to.

leesalminen 17 days ago [-]

Just an anecdote for you- in 2009, I wanted to drop out of college because I preferred partying. I obviously needed income and was dissatisfied with the jobs I'd had in the past (fast food, waiting tables and hotel customer service). Like you, I had always been a tech "nerd" and had built websites and recycled computers for my church/charities in the past. One day I came across a job post for a small SEO firm that was looking for a "web developer" aka clicking around Wordpress, writing some CSS, and writing SEO content. It paid \$11/hr. I sent an email and he agreed to interview me. I must've charmed him because he agreed to give me a 90 day trial. It worked out and I worked for him for about 2 years before a recruiter found me and got me a job at a "real" software company on a team of 15 engineers. After a couple years there, I founded my own SaaS side business (which went viral on HN circa 2014). It grew to become much, much more than a side business and was eventually acquired by a PE firm.

I just wanted to say to keep your head up, remain persistent, keep learning and stay curious. It can be done and hard work (usually) pays off.

Stay flexible - even if a job doesn't seem super relevant to your core interests you never know when it'll come in handy later. For example, my experience in fast food gave me great insights into POS systems (which I ended up building) and my time working at a luxury boutique hotel taught me how to provide excellent customer service even in difficult situations (one time I didn't call a customer's room at the request wake up call time and he missed his flight- eek).

Best of luck to you!

Ps- just sent you an email

mkoryak 17 days ago [-]

But were you able to keep partying? And for how long and how hard?

leesalminen 17 days ago [-]

Hahahah, the most important question! I was a fairly prolific partier until about 2014, right when I started building Gingr. I actually started building the app while on partying-related probation. The inability to party hard was what drove me to code after-hours. Today, I'm pretty lame at least according to my wife and son. Thanks for asking ;).

vcanales 16 days ago [-]

Just wanted to say that I read Grindr and I had to do a double take :)

xupybd 17 days ago [-]

You've just tipped me over the line. I'm finally going to start my side project.

leesalminen 17 days ago [-]

Best of luck! Remember- what's the worst that can happen? Even if it doesn't become a commercial success, surely you've learned something(s) new that you can use later. Hope to see it on Show HN one day!

scomp 17 days ago [-]

Side question sorry, what happened to the side business after the PE firm acquired it?

leesalminen 17 days ago [-]

It's still going strong today! It's led by my "protege" and they have 40+ employees now with 8-digit revenue. They just acquired a second competitor in our space. I couldn't be more proud of them. Check out www.gingrapp.com.

ux-app 17 days ago [-]

I don't want to derail the conversation, but I'd love to hear more about gingrapp. What made you choose this niche, what were the early days like?

leesalminen 17 days ago [-]

Well, I had recently adopted an Australian Cattle Dog puppy but had a full time in-office job. They're a very active breed and leaving her at home all day wasn't an option. That prompted me to find a dog daycare, something I had never heard of (and honestly thought was slightly ridiculous). When checking out the first day, I noticed their business was a mess. Everything on paper, in messy stacks at the front desk. And then it took almost 5 minutes for them to be able to accept my card payment (one of my biggest pet peeves). That got me thinking- why isn't there software in this niche? It turns out there was, but it was all CD-ROM software straight out of Windows 98 land. I didn't know Jack shit about dog kennels so I emailed 10 local businesses asking if they'd beta test what I built and provide feedback. 1 wrote back so I drove over that day. The owner didn't know anything about computers but he set me up with his manager- an aspiring product designer. We hit it off immediately and started collaborating. Eventually we became 50/50 business partners. Once we had built something we decided to show it off at a trade show. Turns out there was one in our city the following week! We bought a booth, and a shitty sign and 2 polo shirts and showed up. Every single person there lined up to see what we had built. We knew we had something. Got our first customer that day. Eventually it became more work than we could handle on nights / weekends, so I decided to pay my partner's bills out of my own pocket (from my day job) so he could quit his and run the day to day until we had revenue to pay both our bills. We got there after a few months. After that, we worked 16 hours, 7 days a week for about a year. We started hiring people. The rest is history, as they say.

To this day, my partner is my closest and most trusted friend, even though we haven't worked together in 2 years.

Dunno what else to share, feel free to ask!

Dragonai 17 days ago [-]

Absolutely love this. Proof that opportunity is everywhere. Congratulations on an awesome success story!

hikarudo 17 days ago [-]

What an amazing story! Thanks for sharing.

LegitShady 17 days ago [-]

congrats and thanks for sharing

40four 16 days ago [-]

I have used Ginger! There is a Pet Paradise in my city we used to go to, and I booked through the app all the time. Small world haha

leesalminen 16 days ago [-]

Oh, awesome! We love Pet Paradise- they run a good business and the leadership really does care about pet safety.

40four 16 days ago [-]

Probably the nicest daycare/ boarding brick & mortar facility I've ever seen. Great people working there too. We only stopped going because we moved to the other side of town and it's too far, there is another place closer.

lynguist 16 days ago [-]

But how did you pay for your livelihood at 11\$/h? That costs in the end more than what you make! I was in a similar situation, but I quit because the job was costing me more money (paying for transit and food) than I was earning.

lukasdanin 17 days ago [-]

Thanks for sharing this story!

leesalminen 17 days ago [-]

Hopefully it helps motivate someone. I'm pretty astonished with how my life turned out, seeing that a partying college dropout's trajectory isn't exactly pointed towards the moon.

Bakary 17 days ago [-]

If you don't mind me asking, what prompted the change in mindset from wanting to drop out to party to working hard and ambitiously?

tomcam 17 days ago [-]

Lovely story!

playingchanges 17 days ago [-]

One more angle since I haven't seen anyone mention it.

Instructions on how to bootstrap a software portfolio:

Pick a language (sounds like you chose JS which would be my pick as well), buy a copy of cracking the coding interview, make a leetcode.com account, make a codepen.io account, and get to work.

If you're smart and dedicated you can teach yourself this stuff and these are the best tools to help you in my opinion.

Spend your time solving problems on leetcode and then utilizing these techniques in codepen portfolio pieces.

In my opinion with serious dedication you can have a junior swe worthy resume and portfolio put together within a year.

[edit] since the question inevitably comes up with JS in my opinion you should not spend any time focusing on front end frameworks. Learn Vanilla JS, HTML and CSS, you will blow your interviewers away if you can solve their problems without a framework and it is overhead you don't need as a beginner.

_drimzy 17 days ago [-]

> buy a copy of cracking the coding interview, make a leetcode.com account

In my opinion this is a pretty bad advice and I see a lot of entry level programmers struggling for a long time because of this. A basic algorithms and data structure book (something like Introduction to Algorithms by Cormen, and Data Structures by Mark A Weiss) is a must before jumping into leetcode/cracking the coding interview. One needs to have a foundation before diving straight to interview problems.

oatmeal_croc 17 days ago [-]

Also, having a Cormen handy is always a good idea regardless of your expertise level with data structures/algorithms!

beezechurger 17 days ago [-]

why cant ctcI be supplemented by youtube videos? perfection is our enemy and many non faang jobs dont do LC interviews

_drimzy 17 days ago [-]

Because ctcI is a terrible book to start learning basics from. I agree with let perfect not be the enemy of the good, but CTCI/LC are not good, they are terrible and overwhelming for someone with no background

in algorithms.

catwind7 17 days ago [-]

agree with this ^. Pretty sure the book was written with STEM (but really, comp sci) new grads in mind

onion2k 17 days ago [-]

Cracking the Coding Interview and leetcode are good advice for getting a job at a FAANG or a startup (not that startups hire juniors.), but they're much less relevant at small- and mid-sized software companies. Applying to a FAANG/startup you're up against top graduates, people who have been coding for decades (even for junior roles), and *highly* motivated people who already have more demonstrable experience than you.

If you're applying to somewhere small it's less important to demonstrate you're already a good developer, and more important to demonstrate that you're interested in learning what the company can teach you, you're a decent human being who the company will want to be part of the team, and that you're not going to give up after a few weeks. Tools like leetcode won't teach you those things.

kitsune_ 16 days ago [-]

Agreed. I know plenty of highly intelligent, super talented and experienced people who had to take multiple rounds to get into Google. I myself tried it once and made it to the hiring committee, but was ultimately rejected. I was an experienced developer and prepared myself pretty well and was still nervous as hell.

To be honest I don't know how without any prior experience or graduating from a top uni you would get an interview at a FAANG in the first place.

kitsune_ 16 days ago [-]

I think this is partially bad advice, sorry.

Sure, algorithmic thinking is valuable, but linear programming and complexity theory has no bearing on 99% of all web development jobs that require JavaScript and in this context I think this fetishisation of Informatics Olympiad style puzzles is the wrong advice for someone trying to enter the field. You do not need an in-depth understanding of computer science to get started with programming or survive in the field, especially not in an enterprise environment. I met Java developers who I suspected didn't even know what class is to be honest. But people like this can thrive at big corps.

In the first year of university I had Prolog, EBNF and partial derivatives, theoretical foundations of computing, linear algebra and stochastics but how much of that stuff do you really need when programming a UI? The complexity lies elsewhere, understanding business requirements, talking to people, avoiding technical debt by talking PMs out of weird requirements, structuring your application in a modular fashion, staying up to date with the eco-system etc.

I agree that focusing on HTML, CSS and vanilla Javascript before jumping into React, Vue or whatever is a good idea tho. For frontend development job, the best thing in my view would be a solid understanding of the web fundamentals + experience in one of the big three frameworks. Outside of that, I think there are probably a lot of jobs where being able to just work with Wordpress templates is good enough tbh.

Anyways, when I'm hiring "juniors" or I'm looking for something that tells me the person a) actually likes programming b) has some demonstrated talent (problem solving, compositional thinking, whatever you want to call it) and is able and willing to learn and c) is able to work together with people (which is the biggest problem with self-taught solo-devs, not having worked in teams before) d) is not an asshole

divbzero 17 days ago [-]

> ... *how to bootstrap a software portfolio* ...

This is how I got my first job in the industry. I built software and websites to solve problems that I personally found useful. Though the solutions were (in retrospect) rather unpolished, they were good enough that a team saw some potential and were generous enough to take a chance on me.

broodbucket 17 days ago [-]

Initiative and enthusiasm are pretty big factors that personal projects demonstrate, even if the actual outcome isn't anything to phone home about. If you're genuinely interested in the type of thing a job requires, you'll naturally pick it up faster.

blackrock 16 days ago [-]

L33T Code teaches you nothing about building reliable, fault-tolerant, and performant computer systems.

It's irritating as fck that these FANG companies are so brain dead, that this is all that they can focus on.

agumonkey 16 days ago [-]

What annoys me is that demos are not really solid proof. You may write something functional but full of yet to find bugs and bad style, perf issues.. where do you stop, how polished and solid should a portfolio be ? I'm on the perfectionist / obsessive kind and its really difficult for me to balance that.

gabereiser 17 days ago [-]

This is the best advice. It's 1000x easier to learn this stuff remotely than it was when I was starting out in the late 90s. Back then it was all black magic or institutional knowledge or higher-education backed.

Learning JS would be my pick too. In fact, there are coding schools/bootcamps that also make this their #1 pick.

Learn JS, HTML, CSS. Play on codepen, make things. Once you've got that down then you're ready to tackle cracking the coding interview and get that job.

It's hard for us vets too you know. With ever increasing requirements, frameworks, tools to learn. We also practice this *make stuff* in order to stay *relevant*. Coding is a life-time of learning (new frameworks, new languages, new ways of doing things) and is very rewarding. Graphic Design is as well in the right brain category of work. Web needs both. Right brained people, left brained people, new coders to grow teams, old coders to mentor and lead those teams.

Either way, either path you choose - Backend or Frontend, there's never been a better time to learn this from online resources in the comfort of your PJs.

wishinghand 17 days ago [-]

Suggesting "Cracking the Code Interview" for a front end dev should come with this repo, where it has solutions in Javascript: <https://github.com/careercup/CtCI-6th-Edition-JavaScript>

iainctduncan 17 days ago [-]

Applying for remote work is a *sales* job. I have done it a bunch of times, landing work within two weeks each time, and this is how you do it:

- figure out for whom you are a good fit based on your experience - for whom and what are you a *good* solution?
- create a professional, to the point resume, highlighting your unique value prop for these people, and your experience that they will care about
- write a really good cover letter about it, and say in it "I realize you may not have considered remote, but I think I am a very good fit and would be a benefit to your company, would you consider discussing this possibility"
- go through stack overflow careers making a big list of *everyone* you'd be a fit for, *regardless* of whether they think they will hire remote. The whole world is your oyster.
- send out applications TWENTY TIMES A DAY. <--- THIS IS WHAT MATTERS
- follow up: after a few days, after a week, after two weeks, then drop them if you haven't heard.

That's really all there is to it, it's a sales grind. You get up and you do your TWENTY calls every damned day and you will find work.

Some folks might think these are high numbers, but here's the rub: when the answer come in, you want *options*. There's a sales saying: the best negotiating tool is a fat pipeline. If you get a hundred applications out in a week, and they take a few days to a week to get around to them, the following week you have a bunch to look into. But don't snooze, keep firing out another hundred that week until you close! That way, you get not just a job, but one that is actually good for you. You can do it, it's a sellers market in tech! So grind the numbers. Good luck! :-)

charleyramm 16 days ago [-]

I needed this advise myself a few years ago before landing a similar role. What helped me was claiming 'job seekers allowance' which comes with a requirement to grind out a certain number of applications per week.

virtuous_signal 17 days ago [-]

As a self taught graduate in an unrelated major, in a rural area, I was fortunate to get an entry level job and then a fully remote job this past year. If you can do any contract work or part time work in the field to start with, that would help. Welfare helps. Then I would try to make sure not to target FAANG types of jobs or any markets where every job posting gets 300 bootcamp applicants. Some concrete advice below:

I learned javascript + basic web development stuff like everyone else, because that's where the most beginner resources are. But the job offers I actually got were due to my learning Java (with some Java enterprise edition mixed in) and having some book knowledge about it. There are vast swaths of industries that will help you enter the middle class, and then some, by working on their old Java or C# applications. A ton of career advice out there is targeted towards the 1% who are shooting for Silicon Valley. We don't need that.

Also try to get proficient at Leetcode and come off as intelligent. A lot of your competition is computer science majors.

Lastly, and this might involve some conscious or unconscious deception: Make it sound like you intend to move to wherever the job location is, post-COVID. This works better if you are already in that state. Once you land the first job, maybe they'll

offer the remote option eventually; otherwise keep applying to a couple jobs per day; this job search will be slightly easier and you will actually have the leverage to ask for a remote option when it comes up.

whoknew1122 17 days ago [-]

> "Then I would try to make sure not to target FAANG types of jobs or any markets where every job posting gets 300 bootcamp applicants."

The second job I landed in tech was a FAANG. I completely agree that you shouldn't target FAANG type jobs for your first job. Not because of the other applicants. I could care less about the other applicants.

Where I work and do tech interviews (AWS), it's not about the other applicants. We don't look for the best person out of a pool of applicants. There's one question and one question only: Is this person better than half of the people currently doing this job.

If you're better than half of the people currently doing the job, you're hired. If not, you're not. And the issue is that it's really hard to have the breadth of knowledge necessary to meet that bar without previous experience.

I work in premium support for security. To be better than half the people already here you have to know the following really well: Linux or Windows; Networking; DNS; Encryption; SSL/TLS; Network/OS Troubleshooting; Web App Vulnerabilities; DDoS attacks and mitigation; and more. It'd be very hard for anyone who hasn't done this professionally to be exposed to enough tech to have that sort of depth.

msg 17 days ago [-]

I interviewed people for Amazon as well before moving to another FAANG. All I would amplify here is that entry level is entry level, especially in software. None of the college hires have professional experience (maybe internships, but they had to start from nothing to get those anyway). So what you are trying to do is

1) not be an asshole; look at their leadership principles and figure out how you resonate with them; answer honestly if you never did that

2) beat the technical questions. they are a proxy for skills and experiences

I used and recommend Leetcode for #2... for entry level I don't think you should need the paid tier.

If there were a Leetcode for not being an asshole, I would recommend the paid tier.

whoknew1122 16 days ago [-]

> "I interviewed people for Amazon as well before moving to another FAANG. All I would amplify here is that entry level is entry level, especially in software. None of the college hires have professional experience (maybe internships, but they had to start from nothing to get those anyway)."

100% agree with everything you said, although depending on the Org 'entry level' can mean 'just graduated from college'. In Premium Support, at least, they're always tinkering with their recruiting models.

I was hired as entry level with 1 year experience at a startup. These days, that entry level job is reserved for recent college graduates or people getting out of the military. They wouldn't slot me as entry level today for recruiting/interviewing purposes.

Not sure how AWS handles entry level software engineering positions.

agumonkey 16 days ago [-]

What were the things that surprised you the most coming from outside IT ? what were the hard part ?

sokoloff 17 days ago [-]

Here's my advice which is worth hopefully at least twice what it costs you.

Focus on achieving some initial level of "proof" that you can perform in an entry-level software (or help desk) role. If you can't ever visit the office, you're competing against a worldwide set of candidates, but you can still bring something that distinguishes you. That you work in US time zone and make it easy for someone to employ you are surprisingly strong benefits to an employer who has so far only employed US people.

If you want to break into programming, go after that rather than help desk or sanding body panels. If your next best alternative is moving far away, renting a place there, and doing auto body work for minimum wage, you can probably come out after-everything even by staying put and doing a mix of online learning and even terribly paying gigs on fiverr/ upwork/ etc.

Once you get to the point where you have the basics down and are the equivalent of a boot camp grad, you have more options for full-time employment. Even at bootcamp grad level, most companies are losing money on you for a year, so before that point, it's a really tough sell.

Maybe consider Lambda School as well. They've got a repayment program that scales with income, so if you don't manage to make the turn and break into coding somehow, you're pretty much off the hook after some time. (Obviously read their terms, don't rely on my summary.)

I applaud your focus on your family; I'd keep that out of the interview process. If I'm hiring an entry-level remote employee, I don't want to worry that they're taking care of a family member most of the time and trying to fit my work into the gaps. If that's what is happening invisibly behind the scenes, great, but it's irrelevant or negative to the interview.

"I have strong ties here and am only open to remote work" is all I need to know as an interviewer.

Best of luck; once you hit the "I am at a boot camp grad level or better", we have remote-only positions available (as do many companies). Provided you're in a state where we have the ability to hire, I'd be very happy to have you apply.

Best of luck on your journey!

virtuous_signal 17 days ago [-]

> I applaud your focus on your family; I'd keep that out of the interview process. If I'm hiring an entry-level remote employee, I don't want to worry that they're taking care of a family member most of the time and trying to fit my work into the gaps. If that's what is happening invisibly behind the scenes, great, but it's irrelevant or negative to the interview.

This is really important. Don't think about employers "taking a chance" on you or having their hearts warmed by your story. (I mean, it might actually work when the OP is hitting the top of HN but in general...) They want non-risky choices.

pmiller2 17 days ago [-]

> If you want to break into programming, go after that rather than help desk or sanding body panels. If your next best alternative is moving far away, renting a place there, and doing auto body work for minimum wage, you can probably come out after-everything even by staying put and doing a mix of online learning and even terribly paying gigs on fiverr/ upwork/ etc.

Since a sibling comment already covered the other thing I was going to say, I'm going to focus on this part, as it's 100% correct.

Moving away to take a minimum wage job is most likely a trap. You will not be making enough money to both support yourself and do anything meaningful for your family, and you're going to be physically tired after working 8+ hours a day. It's definitely not a good long term solution, and it seems pretty terrible in the short term, too. Even taking a minimum wage job locally seems like a better solution to me, if at all possible (I did see you mentioned you live in the "poorest region of the United States.")

I think I would start by focusing on what the minimum amount of money you need to make to support your family, then seeing how you can make that much remotely. That can be whatever, because it's just a stepping stone allowing you to move on to finding a remote job that not only lets you take care of your family, but also lets you get ahead.

You've got a hard road ahead of you, OP, but it's one I've been down myself. For the longest time, I never had more than a few hundred bucks to my name. My net worth was deeply negative due to student loans until very recently. Even though I'm in a much better place than that now, it's still been 3 steps forward, 2 steps back for me. At the minimum, you will need a lot of perseverance, a little luck, a dash of chutzpah, and a willingness to keep improving yourself.

rmk 17 days ago [-]

Why not start as a technical writer and then work your way up? People will have ZERO qualms about hiring a remote technical writer. Writing technical material will also improve your understanding of technology and give you an 'in' into the industry.

Another alternative is to join up as a sales person at an enterprise company. Look for companies that are growing at a rapid clip; they will often hire people whose sole qualification is that they have a body temperature of 98.6 F. Sales includes helping people do POCs and such, which is part of the sales process for complex enterprise products. Once you start there, you can work your way into Software Development if you wish and can find helpful colleagues. The advantage of Sales is that they require a regional presence, and there is a chance your location will work out to your advantage (typically enterprise companies have sales forces organised by vertical and/or region, including regions of the United States).

The last option is to join an enterprise company as a Customer Support person. Once again, you can develop basic skills and become a helpful person by trying your hand at scripting to help perform minor tasks support people do when troubleshooting customer issues. I see plenty of bright Customer Service people move into Software Engineering proper after they have demonstrated intelligence and a willingness to help the customer by going a little above and beyond the immediate problem.

Most of my advice is centered on Enterprise companies because that's where a large workforce comprising people with a variety of technical skills can be found. Consumer web or mobile type companies are a bit too narrow in the types of people they hire, and frankly, the variety of interesting coding and problem solving opportunities at such companies pales in comparison with Enterprise shops.

grumpwagon 17 days ago [-]

I'd highly encourage you to continue learning to code. You might also consider software QA. It is also easy to do remotely, pays well, and the learning curve is a bit lower. If you learned selenium, the ability to navigate a SQL database, postman to

test REST APIs, and some testing terms and methods you'd be well on your way. The first job will be the hardest to get, but once you have that, remote jobs are readily available.

Obviously you didn't lay out all the reasons why you couldn't move, and I don't expect you to, but I'd also REALLY strongly consider the implications of what a steady, well paying job can do for a family. Relocation opens up a world of opportunities for you, which opens up support and services for your family (paid and otherwise) that just aren't available in the poorest places. You don't have to be away from the people you care for if you can bring them along. I understand that leaving a support structure, even a flawed one, may not be possible, but think about that REALLY hard before you rule it out, especially if you're young. Some initial pain may transform lives. There's a reason so many people move away from the places they grow up. Plus, if you move for your first job, you always have the option of moving back after a year or two if it doesn't work out once you've established yourself in the career, which would make finding remote work much easier.

cjohnson318 17 days ago [-]

> pays well, and the learning curve is a bit lower

I've done development and testing, and I would characterize QA as "same job, less pay". In both roles you struggle with vague requirements, lots of edge cases, and unrealistic deadlines. It is surprisingly difficult to get out of the QA pigeonhole. People assume the job is easier, and that you're not as capable as a developer.

rmk 17 days ago [-]

As a developer, I emphatically disagree about the "same job" aspect. QA requires a pretty different mindset, one which developers lack entirely. This is particularly true about testing things like storage appliances, routers, switches, and such. Putting such complex systems through their paces requires specialist QA folks, including loads of manual QA'ing in addition to heavy automated QA. That said, despite how critical QA is to putting out a product that's even remotely serviceable, the market for QA folks generally pays lower. Apparently, if you are putting out reams of code and tossing it over the wall to the hapless QA guy, you deserve to get paid more. Much more.

I have the misfortune of working at a place where some "bright" person got the bright idea of "eliminating" the QA budget because "developers should be able to do everything a QA can". QA was dissolved and existing QA specialists were told that they had to learn how to program, or get lost. The product quality went down the tube and the place got grief from (Fortune-500) customers who write fat checks that run into the millions for support alone. That has been a train wreck in slow motion.

The simple fact is that you can *maybe* get away without separate QA staff for a something like an internal website, but for anything else, assuming that developers will test their work and put enough thought into even the most common use-cases and certify that the system as a whole works is delusional.

mdip 17 days ago [-]

I mostly agree with your statement and your issue with companies wishing to eliminate the QA budget (never a good idea).

But I question this bit: "QA Requires a pretty different mindset, one which developers lack entirely."

I don't think that's true. I've done both roles, successfully, at various times. The key, I've found, is that a developer can't test their own work. Developers of a project should also not be put in charge of testing each-others' work if they are working in close capacities. A QA strategy that begins and ends with "test your own stuff", "code/peer reviews" and "write unit tests" is going to reach a quality cliff as the complexity of the software increases. You can get away with it, sometimes, but it doesn't scale.

I'm not sure what the cause of this is, or if it can be improved (it can, but can it be completely eliminated?). It's a similar phenomenon to proof-reading your own work. You mentally add words that aren't there. You follow happy-paths in your software because you know what they are; you don't "misuse" your UI enough[0].

[0] One I think of often was when I wrote a UI for a desk conferencing device. I tested every aspect of it and it was returned to me shortly thereafter when the QA guy joined/exited about 15 "on demand" meetings as rapidly as possible throwing the UI into a state that couldn't be recovered from without rebooting the device.

rmk 17 days ago [-]

You hit the nail on the head. It takes time and effort to cultivate the QA mindset, and to shift into it from a developer mindset. Doing both simultaneously is often difficult in practice because it increases the lead time to produce the end-product, something that Project Managers don't seem to grok. You can successfully do both if you alternate between them, spending substantial chunks of time immersed in one role or the other at any given time.

Developers can be reasonably objective when it comes to writing small, self-contained unit tests and some amount of test automation, but beyond that, where serious QA begins, developers are generally too much in love with their creations to distance themselves from them and view them objectively.

You can see this pattern with authors who become (commercially) successful. The publisher, whose most valuable service is editing the author's work, starts becoming too deferential, causing the author's work to become muddled and flabby. A good editor is often the only one standing between mediocrity and greatness...

Cyph0n 17 days ago [-]

> This is particularly true about testing things like storage appliances, routers, switches, and such.

Yep, QA is a must for complex systems.

I write for an OS that runs on a service provider router and can assure you that no dev team can come close to understanding how to approach testing the entire product. So QA becomes invaluable when it's time to run system-level (integration) tests that tie multiple features together.

Of course, there is always a lot of stuff you can automate at the system level. I think the QA and test teams do a good job of identifying these areas, though.

Igelau 17 days ago [-]

> QA requires a pretty different mindset, one which developers lack entirely.

This place cracks me up sometimes :)

sokoloff 17 days ago [-]

QA and dev are not the same job IMO/IME. Simplified:

QA is "the system should do this; *does* it do it?"

Dev is "the system should do this; *make* it do it."

I'm not putting one above the other as both functions are needed for most successful companies. I *am* saying they're not the same job.

blackrock 16 days ago [-]

Not exactly. QA is all about breaking shit.

It's already assumed that the product has certain features. And that those features work a certain way. But what if you did something else, that was unexpected. Then does it break the system?

As a QA Engineer, your job is to find that fault, and break it. Because you are possibly the last line of defense, before that software goes out into the real world.

munk-a 17 days ago [-]

I disagree in that I think neither position actually gets a "the system should do this" - Devs will put together a solution they think will fulfill a user request and QA has the difficult job of figuring out why-in-the-hell the dev thinks that's something a user will figure out. /s

More seriously there is a lot of differences between the two tracks but a QA person who can write clear and repeatable reproduction steps will save a company a whole bunch of dev hours and doing so usually requires a pretty good understanding of why the system is actually breaking.

isv 17 days ago [-]

I think in many cases the QA job is harder. The developer needs to implement the feature according to AC and cover known error conditions. The QA needs to test the feature according to AC and also throw everything under the sun at it to uncover unknown error conditions.

However it's been a sad reality from my experience in the industry that the average QA engineer is a less capable coder than the average developer. There could be many self-reinforcing reasons for that which includes lower pay. There are few things good developers wish for more than equal or superior QA engineers who make developers' job writing quality code much easier.

anm89 17 days ago [-]

Just to give an alternate take, I think you can get pigeonholed as a QA person and if you want to be a developer it may be a better route to look for positions that are going to give you practice developing.

That being said, the reality of job market these days might be that entry level positions are too competitive and this just isn't practical in which case getting your foot in the door with a QA role is still a much better option than nothing. But I would say it is a fairly indirect route to where you want to go.

littleninja 17 days ago [-]

As someone who came into software dev without a degree and started in QA, this is correct. Some shops are setup with QA as a separate function on separate teams. If development tickles your fancy, starting on one of

those QA teams may limit you.

Sharing from my anecdotal experience, if you'd prefer an environment where you can grow beyond manual testing (QA Engineer) and automated testing (SDET), look for positive cues: "QA as a role" or cross-functional teams or embedded QA within development teams. Ask for stories (examples) about a team member who made a lateral move, how leadership looks for this potential and supports the transition.

curiousllama 17 days ago [-]

QA is a VERY solid career, by any general standard. OP is looking for a job, not a creative outlet. Shouldn't apply an SF lens to this.

grumpwagon 17 days ago [-]

Totally agree you can get pigeonholed, but my read of the OP is that they want a job that can support their family first, and that development was one of the things they were considering, not that development was a goal in and of itself. My thought is that QA is a solid career, and that once they are established, and get a support structure for their family, if they want to become a dev they can work towards that from a position of strength, rather than a position of near-crisis.

sokoloff 17 days ago [-]

Plus, it's 1000x better than sanding Bondo all day!

faldore 17 days ago [-]

I went this route. Once you are in QA it is extremely difficult to get out. Everybody views you as "a tester" no matter how good your development skills are. If you genuinely want to be in QA as your career, or if you are truly desperate for work and willing to spend the 5 years or so it will take for you to dig yourself out of QA, then do it. It worked for me. But if you really want to dev, do not take a QA job unless you really can't find a dev job.

pmiller2 17 days ago [-]

100% correct. I did a brief stint in QA before I moved into dev. That job no longer appears on my resume, because I don't want to do those things anymore.

I know a guy who's a senior QA and has been writing automated tests in Python for over 10 years. He can't get hired as a developer because of this "pigeonhole effect." It's nothing to do with his personality, either. He gets hired in QA with no issues.

asidiali 17 days ago [-]

This a great comment, and solid feedback. QA sounds like a great step for this person.

samtheprogram 17 days ago [-]

I've been on projects with non-developer (but technical) QA that would barely go as far as diving into development tools... the pay for those team members was extremely low, but they were foreigners and the pay was still above minimum wage. If you wanted to be pedantic, these were, simply put, just testers.

With that in mind, I think OP immediately has a shot on job boards for such a position if priced correctly — especially with the added perk of great written English / better communication.

EDIT: that's not to say they should not learn to code; agree with your general sentiment, but I think growing in that sense shouldn't be a barrier to entry.

exdsq 17 days ago [-]

Remote support roles will probably be your best bet, perhaps for a startup where it's focused more customer experience than tech skills. Do that while carrying on with courses and you'll be in demand quickly!

Regarding finding work, I'd go on remote-focused job boards and look for any junior support or QA roles. Email the hiring team directly saying hi along with your CV to try beat any filtering systems. If you can't find many of these, or work through all the ones you can find, try messaging small companies hiring onsite and see if they're up for remote work anyway.

Persistence goes a long way, expect to be told no a load of times but I hope you'll be able to land something. Good luck!

marcusverus 17 days ago [-]

I agree with this assessment. I'd also recommend searching Dice and Indeed for something like this.

It's also important to do a bit of research and understand the huge variety of jobs which will come with titles like 'Remote Support' or 'Remote Support Analyst'. An IT Helpdesk job, a Software Application Support job, and a Zappos customer service job might all share the exact same generic title. So do some googling to figure out what is what, make yourself a list of titles that match the jobs you're looking for, then troll the job boards for roles with that title. Based on the post, I would target jobs with titles like "Application Support Analyst". If you search for remote only with no geographic constraints, you will find an absolute ton of jobs.

As a former recruiter, I think it's worth noting that job hunting in this way is absolutely a numbers game. When I was looking for IT work with little/no experience, I would generally get my first interview after ~50 applications. My point is that you shouldn't get your hopes up with each application, and you shouldn't feel let down if you don't hear back on a given job. If you go in knowing it will be a bit of grind, it will be much easier on you. Set yourself a goal of 2-3 applications a day, and get that many done, every single day. Given the huge number of fully remote roles out there, I'm confident that you would be hard pressed to apply for 100 of them without getting an offer.

With that in mind, get your resume all set, write out a couple of cover letters (your writing is solid, so this will help. But don't write an original letter for each application.) and then apply for *everything* that looks like a fit. Ignore experience requirements for roles that aren't "senior" or "manager" roles.

Best of luck!

germinalphrase 17 days ago [-]

It is generally easy to identify, find contact information for hiring teams?

exdsq 17 days ago [-]

So I was specifically talking about small companies, but ultimately I think so. I'd recommend sending it to an alias address if you can find one, but ultimately LinkedIn and firstname@domain will get you far.

vinger 17 days ago [-]

The easier it is the harder this technique becomes.

pfarrell 17 days ago [-]

As someone who has sat on the interviewer side for some entry-level positions, I would recommend you do not present a bootcamp project as an example of your work unless you have greatly extended it. I have sat on many interviews where I had the exact same project shown to me. It began to have a negative signal unless the interviewee had pushed beyond the requirements.

I would recommend that you develop some project on your own which you can speak to and talk about struggles with. Depending on the position and the company, demonstrating and talking about personal experiences on a project can go a long way towards showing your passion and determination. This isn't the way in to every company, but it is a way to ensure you have something to talk about that will help you stand out during an interview.

curiousllama 17 days ago [-]

I like some of the advice in this thread, but taking a bit of a broader view:

OP, you're trying to do 2 things here: (1) get a job, and (2) learn a trade. Those are different, though related pursuits.

The ideal is you can find a job that will teach you the trade; in other words, you're looking for an apprenticeship. There are often gov programs for tech apprenticeships (typically state level, so I can't link a definitely-relevant one), but [1] may be a place to start. Also just google "[state] IT Apprenticeship." There are also corporate programs at IBM and Accenture you should look at.

Second, even if you can't get the tech job immediately/this round, don't give up. Google needs IT support folks enough that they started a new education program to teach people enough to work for them [2]. If you have to get a different job, look at that. It is a bit pricey at \$49/month, but Google says they'd consider completion to be equivalent to a bachelors.

Finally, every CV you submit should have a cover letter, consisting of the paragraphs you wrote in this post from "I am a techie at heart" to "solution seems impossible."

Good luck!

[1] <https://www.apprenticeship.gov/apprenticeship-industries/inf...>

[2] <https://www.coursera.org/professional-certificates/google-it...>

hosh 17 days ago [-]

I know this option isn't the best, but check out rentacoder or freelancer. You're going to get underpaid, and competing with lowest-cost vendors. This is contract, 1099 work, so no employee benefits. (But! If you play your cards right and understand taxes and laws, it can be more lucrative).

However, what you have going for you is that you can speak and write English well. You do not have to (and should not mention) where you live, or your own circumstance. Most people are looking for someone cheap to do short projects.

The goal here is to bootstrap your experience with paid projects, even if the pay is low. You use that to then get in front of larger projects or part time work.

Pick a tech platform or a platform family and stick with it, at least initially. You're looking for projects that, after completing them, will get you in the door with stuff that pays well.

Anything that you develop that you can put into github, you should. Usually, contracts are done as work-for-hire. You might be able to negotiate keeping copyright. You might not. However, tools and scripts you write to assist with it should get

posted. Your hobby projects should go there too.

Now, to talk specifically about your strengths. If you love teaching and you are good at writing, one thing you can do is to start blogging about your projects, or writing tutorials. It is a kind of psychological or marketing jujutsu. Teaching something implies expertise and authority, so long as the content is competent. You can even frame this as documenting your experience learning something. Just make sure that you brand this as you authoring the thing.

Last, give some thought to speciality. If you are known to be able to solve problems in specific areas, people come to you. Some people write web apps. Some people do devops/sre/infrastructure. Some people write low latency multiplayer game servers.

karaterobot 17 days ago [-]

OP, this is the most realistic advice for your very short time frame.

Take on a freelance contract and kick butt on it while simultaneously looking for the next contract. Eventually you will either get offered a position by one of your clients, or you'll have enough of a resume to start applying in other places (or you'll like freelancing enough to stick with it).

tkgally 17 days ago [-]

I was going to suggest the same thing: While continuing to look for steady work, see if you can get freelance jobs through any of the online boards. I started out as a freelancer thirty-five years ago (in another industry) and I'm happy I did: though I now work for a salary, that freelancing experience laid the foundation for my current career. And freelancing remotely is much easier now than it was thirty-five years ago.

A couple of years ago, I farmed out some small web design, programming, and proofreading jobs through Fiverr. The price competition among the people who bid for the jobs was brutal—people were bidding from some of the lowest-income countries in the world and seemed desperate for work—but it soon became clear that I would just be wasting my time if I went with the lowest bidders. Your writing skills will do a lot to attract jobs at decent pay. Good luck!

hosh 17 days ago [-]

(So don't just do documentation pull requests. Blog about the project itself, why you like it, help promote it, write a short tutorial about it on your own blog.)

mceoin 17 days ago [-]

Hi Anon,

Since you have limited time to find a remote job and nobody has mentioned it yet: HTML/CSS still pays pretty well, is fast to learn, and may be even easier to get paid for than Javascript or other coding work since a lot of "better developers" do not want to do it. I would start by building yourself a personal website and using that as your portfolio, and then expand that portfolio as quickly as you can while hustling for work.

On the hustle side, if you're looking for an entry level job in tech that you can do remotely and doesn't require much experience, you can also use Google Maps and Google to get lists of consulting shops in every major city in America (dev, design, marketing, etc.) and just work your way down the list until you have sent a *brief* email to every single one of them letting them know that you're looking for work.

Plenty of other techniques in this thread as well.

Getting your foot in the door is the hardest part. Good luck!

mrzool 17 days ago [-]

I spent months trying to find an HTML/CSS job without success. Every web dev position I found was heavily JS-oriented.

pbreit 17 days ago [-]

The boards are pretty good.

<https://weworkremotely.com/>

<https://www.tecla.io/>

<https://remote.co/>

<https://www.flexjobs.com/>

And if you're up for it, I highly encourage trying to try Zapier which is an amazing company and has terrific remote recruitment. Even if you don't get the job, great experience.

<https://zapier.com/jobs/>

nec4b 16 days ago [-]

> And if you're up for it, I highly encourage trying to try Zapier which is an amazing company and has terrific remote recruitment. Even if you don't get the job, great experience.

I've just check their job openings. In their application form they are asking questions about race, ethnicity, gender,... Is it legal to ask such questions and what is the relevance of such questions?

CptFribble 15 days ago [-]

The reason they ask is that companies are obligated to hire without regard for race, gender, etc - but in the past, many companies would exclude minority groups on purpose but just say "oh we just pick the best candidates, not our fault they're all of the same race."

So the way we are trying to solve this problem is by requiring all companies to collect demographic information on their candidates, and store it somewhere. No one involved with the hiring (theoretically) will see your answers, they just go into a drawer somewhere.

Down the road, the government will ask the company to pull out all the records they collected and look at them in aggregate. If the company is not hiring a diverse enough workforce, there are supposed to be some kind of consequences.

So essentially, companies are required to ask, but you are not required to answer. There will always be an option to say "I prefer not to answer" or something.

nec4b 13 days ago [-]

Thank you for the detailed explanation. Just feels surreal how people are being sorted into groups like some animals based on their looks and ancestry in a first world country.

[reply](#)

throwaway5752 17 days ago [-]

Apply to a lot of them, there is a statistical element to this. Apply to smaller, non-tech companies where competition is not as high. Make your resume as good as possible while being honest.

This may be overly personal, but I would consider contract/temp work, too, to build up your professional experience. That would also reduce the risk of losing your existing full-time job and ending up in a financial hardship.

Also, apply to non-developer roles. There is a bit of complexity to working in a software company, more than just writing code. Consider a support or other development adjacent field that is still technical.

asidiali 17 days ago [-]

Do you have any QA experience?

Feel free to shoot me an email with your resume or Github/LinkedIn, and I'll send back a few openings I know of right now if it seems a good fit. My email is in my profile.

Wishing you well on your job hunt!

AskHNremote2021 17 days ago [-]

I'll send you an email with my real email. I do not have QA experience (unless you count checking items in beauty shops), but I learn fast. We might be a good fit. Sending now.

gwbas1c 16 days ago [-]

Is your family being fair to you?

(I already gave a career-focused response in this thread.)

Can you send one of your siblings to live with your brother? Can you move with the other sibling?

What if you took a job near your brother, and all four of you lived together?

If you move with one of your siblings, will the situation be better for the sibling? (Better schools, better special needs programs?)

Can you and your brother send money back to your parents or another caregiver?

Whenever I hear an "I live in a poor area" story, I always suspect that the situation is more unreasonable than recognized.

Is this whole situation to keep your family farm afloat? Remember, a farm is a business. It's not fair to you to have this kind of impact to your life if your parents (or whoever is running the farm) are working dawn 'til dusk on a failing business.

f1gm3nt 17 days ago [-]

I'm self taught and my story is similar, however in my day, there was no bootcamps. I busted my ass at hackathons to get noticed and moved up from there. I'm a high school and college drop out

When I'm hiring for positions I cross out the school or bootcamp "education". My top priority is to see what you've done in the past and what you're passionate about. If I like a resume I want a code sample, most have their GitHub in it. I review the code and see how well it's formatted and how it's organized. It takes awhile, but I think it's worth it.

So, keep at it, do some small projects on GitHub you're proud of. Bonus if you can find a project and contribute to it, even documentation counts in my book. Mentors are also helpful.

You got this!

timeyyy2 17 days ago [-]

Send me an email. Cannot say for sure in the USA but there is a lot of remote work happening where I am at. If you match our values and have the right mindset there is a good chance we will take you. Timothy.eichler@thoughtworks.com

k_size 17 days ago [-]

There's a company called GitStart (previously known as Murcul). They're from YC S19. I think they mostly hire people to work remotely.

They provide software development service. I'm using their service right now. They seem to provide good mentorship to junior devs, they also have project/product manager positions, and customer success positions. I've talked to their founder and CEO, Hamza Zia. He's a nice guy. In general, they are pleasant people to work with, which means you'll also find pleasant (remote) colleagues there.

all_usernames 16 days ago [-]

I'll give some nontechnical advice: make real human connections. Let people know you (as you have in this post) and be genuinely interested in their work and problems.

Throughout my career, from the first paid job I was offered to the best and most lucrative unsolicited offers I've received in senior management, it's all come down to personal connections. People that knew me personally and knew a little about my life, or secondary connections -- friends of friends.

Your location shouldn't make much difference these days -- everything is happening online. Ask around about meetups, panels, pair programming, clubhouse rooms, or anything else interactive where you can share a bit of your story.

I'm connected to some highly successful people, and their main secret sauce is connections. It's nothing mysterious, but really just basic human kindness, generosity, and a collaborative spirit goes a long way. I find most people genuinely want to be helpful and especially so for people who have achieved some level of success already. So the other part of the advice is to scan your world for accomplished people, and just tell them your story. (Better yet, ask what their problems are and offer to solve something for them.)

This was a great Ask HN, thanks for taking the chance.

grahamburger 17 days ago [-]

I have some extra work that I'd be happy to offload. Probably not enough for full time yet, but I could pay hourly much better than min wage for moderately technical work (like: configuring SaaS products, making presentations on technical projects, writing up project proposals, maybe setting up some networking gear.) All remote, with maybe the opportunity to travel a bit if things work out and you want to. Email in my profile, hit me up if you'd like. Open to anyone interested.

gwbasic 17 days ago [-]

I work remote.

First: Employers hire people who can do the job, not sob stories. Focus on what you can do for a company, and only reveal your personal life situation when required. I have a wife that works odd hours and kids, but I will only bring it up in an interview when discussing how often I will need to travel to an office. (Often I'll just say that I can't move.)

Second: I have almost 20 years experience and I had to apply to about 60 well-chosen jobs before I found a good match. Is "your job is finding a job?" Are you spending 20-40 hours a week looking? Are you choosing where you apply carefully? Anticipate applying for 100 positions.

I suspect you're 35-45 years old. Why are you looking for an entry-level job? Entry-level is also about life experience. You're making a career change, and your existing experience puts you beyond entry level.

But more importantly, "entry-level" implies that your boss will baby-sit you. This is harder to do remotely.

IMO: Focus on "soft skills" jobs like support, direct customer interaction, content. Focus on jobs where you won't need handholding (babysitting) once you know the ropes, and apply for at least 3 jobs a week.

gwbasic 17 days ago [-]

I need to point something else out:

We (my wife and I) chose our situation. We chose to live where I'd have to work remotely. We chose that I would be responsible for balancing a career and childcare. We chose a lifestyle that requires two professional careers.

Why do I bring this up? Think back to when you decided to accept your situation as caregiver to your loved ones. What about your decision not to move. They were choices, even though they didn't feel like it at the time. (Did I have to marry and have kids?) That's why it's not relevant to your job search, beyond a simple "I can't move" phrase.

Everybody who makes good life choices has a situation that locks them down.

29athrowaway 17 days ago [-]

My tips...

Part I: Job postings

- 1) Collect job postings for jobs you are interested in.
- 2) Look at the "skills" portion of the job postings.
- 3) Assess your level of proficiency in those skills.
- 4) Practice each skill until you feel confident you have a working level of proficiency in those skills.

Part II: Study tips

- 1) Take note of every acronym or concept you don't understand. Then, look it up and try to come up with your own definition for it. Write down that definition in your own words. At first, that definition may not be correct, but you have to try to keep it as accurate as you can.
- 2) When you study a concept from a book or website, also take notes. Then, write what you have learned about each topic. Force yourself to explain what you are doing and why.
- 3) Vocalize what you are doing and what you are thinking. Record yourself and then listen to yourself. Expressing your thoughts in a coherent way will help you later during interviews.
- 4) Enroll into a MOOC for Algorithms and Data structures.

Part II: Resume

- 1) Create a 1 page resume that emphasizes your skills and abilities.
- 2) In the absence of formal education, create a portfolio that showcases your abilities.
- 3) If you have completed a MOOC, you can list it there.

Part III: Apply

- 1) Look for referrals among people you know.
- 2) Announce you are looking for a job.
- 3) Apply and interview.

Part IV: Interview

- 1) Don't get too nervous. Stress management is part of the job.
- 2) Budget your time. Questions may have follow up questions. Ask if there are follow up questions in advance.
- 3) Explain your thoughts clearly.
- 4) Ask questions, the interviewer can offer help if needed.
- 5) Learn about the company, what they do, their mission, vision, core values, etc. Crunchbase can help you find basic information about a company.
- 4) Prepare to ask questions. e.g.: what is their engineering organizational structure, what is their culture like, etc. Prepare to answer questions such as why do you want that job, why do you want to work there, etc.

coding-saints 17 days ago [-]

Keep in mind that it's 2021 and yet companies still practice paying wages based on the confidence with each individual when determining offers (more so with software industry). While I encourage you to listen to every comment on here, I also suggest you start matching your learning path in parallel with how you conduct your interviews. Just my two pennies, but if your goal is to dominate the next 2 years to prove your worth, don't forget to provide equal energy and dominate your interviews to gain experience needed for progress. And in case no one mentioned it yet, being a software engineer and interviewing for the position are 2 completely different skill sets. Don't sleep on gaining experience with interviews even for jobs you would never accept. It's not always going to be a fellow peer making the call on your value. It sucks but you'll find out it's not always a straight path to the finish line. Again, just my two pennies...

K2h 17 days ago [-]

Try looking at job postings for 'technical writers' in an area you have a passing knowledge in (computers - and by extension weird industrial systems) and apply for the position even if it's not a junior position and even if it's not remote. These types of jobs can use someone to plow through the minutia, proof read and work on the assignments no one else wants to. When they find out you are only starting at \$5 above minimum wage it should be a really small risk for them to take you on contract to see if things work. I have completely made this up on the spot (sorry) and never seen it work personally, but it is exactly what I would do if I were in your position. When I read your writing above it passes native English, which would be good enough for our team!

update: looks like same ideas as RMK. I should have read the thread farther!

sudhirj 16 days ago [-]

Check out <https://weworkremotely.com/>

WWR is interesting because it has programming and devops jobs, but also customer support and miscellaneous jobs, all remote.

I'd suggest going through the list, short listing the ones that look appealing to you, and then applying the reverse order of desirability, so you get to practice on the interviews you aren't keen on and use them as backups or leverage if you do get through.

I'd suggest starting with something that you can do easily and that will settle you for a while, then you can prep and study for the next-level jobs if you want something more ambitious.

sudhirj 16 days ago [-]

Specifically, I'm suggesting with the remote customer service or community manager jobs while you study and prep for the technical ones. The competition for technical jobs is pretty high right now because lots of people can work remote, and the bar is higher because you can't receive as much on-the-job training. Maybe look for the service jobs with a technical bent of you want a leg up on your tech career path.

seankimdesign 17 days ago [-]

Is your ultimate goal to break into the tech industry? If so, I'd spend a bit more time getting some actual working pieces of code on your GitHub. Editing documentation is a noble deed, but as an interviewer it wouldn't do much for me in assessing your skills.

I know it takes a lot of effort as well as luck in landing that first entry-level job so all I can say is to keep knocking, but also be prepared to show something tangible. Anyone can talk all day about their awesome work ethics but it's another thing to be able to prove it by presenting a polished (for a junior) software.

ARandomerDude 17 days ago [-]

^ This.

I've known several people – usually in their early 20s – who claim to be smart and motivated. Most of them are smart. But when it comes time to actually build something, they'd rather play video games. So how do you distinguish yourself from the video gamers? Build something. Anything.

If you want a full stack job, build a Twitter clone (with limited features) in React (or ReactNative) and Django. Even if you don't use those tools when getting hired, it'll show people that you have what it takes to do the work. Both React(Native) and Django have the added benefits of being in high demand, and extremely well documented.

oehtXRwMkIs 17 days ago [-]

Is it still true that Java (with Spring) is still the best backed stack to know for finding jobs? Or is Python sufficiently popular now for someone to ignore Java?

literallycancer 15 days ago [-]

If you are starting from scratch, just choose the stack with highest average compensation.

fredgrott 17 days ago [-]

Just a suggestion based on personal exp...

I started writing paid articles on medium on developing mobile apps: 1. My first 2 weeks doing articles daily has given me: - 90% of my articles curated by medium staff for inclusion in people's feeds that do not follow me. - inclusion the weekly FlutterForce Newsletter every week. Yes at this point pay is low at about \$5 total but it gets interesting in that it's the same ratio of free value and small paid value as you would do with free features in a mobile and paid features in the same app.

If you already have a small audience to grab somewhere which for you seems to be the case AND if you can make the stimulus checks yet coming stretch IT MIGHT BE THE PERFECT REMOTE FIT.

In my case I know it will even though I have reddit karma of several K in the flutterdev subreddit that it will take me 6 months to get the 500k monthly views on medium that generate on average \$25k and so that plus me self publishing my book in 6 months is what is allow me to leverage to other things.

I cannot promise that you will also have the 500k monthly views on medium in 6 months and \$25k monthly but I think you could get close.

As I get to the 500k monthly I also plan to start developing video course which will put me at the million in views and revenues.

Oh yeah, the best writing tool for dev notes and articles and books is a tool that has been highlighted on HackerNews before namely Zettlr.

If you feel the need to reach out my email is in my HN profile.

wernercd 17 days ago [-]

The crux is you have to have experience to get experience.

I took the first job I could get as a programmer in 2010 at the first place that would accept me... making 21k right after getting my 2 year degree. I now work for a major bank as a Senior Programmer with 10+ years of experience.

If you can afford getting paid under market (minimum wage job you're taking? sounds like thats a yes)? Getting that first job can be worth it in the long run.

Put out a hundred applications and expect 2 interviews. Success is consistency and expectations that you'll have to work that much harder to get that first job.

foolmeonce 17 days ago [-]

I liked eloquent JS, it's a nice introduction particularly compatible with the website debugging style instead of one of the flashy SPAs framework styles.

There's actually quite a lot of piecemeal consulting relating to JS snippets on company websites that might be a better fit than support or more traditional development.

I mean things like e-commerce, SEO, form integrations, cookie consent forms, etc. If you talk to small businesses they will have a strong tendency to use wordpress with plugins to try to manage these things with as little coding as possible.

But often they end up needing a little JS/HTML debugging help and if you look at dev support of companies like Google, Stripe, MailChimp, etc, as you develop skills debugging JS, and try to debug websites then you'll actually be building skills that are normally farmed out by the larger orgs to consultants at pretty good rates.

Similarly, you can also do a lot in spreadsheets, i.e. in JavaScript in Google spreadsheets to do more business related tasks and it can be easier to find small businesses in your area that want to pay for projects to organize their inventory, integrate numbers scraped from different tools, etc.

At any rate, I just thought those might be some ideas that would be a bit different and might have some more local stepping stones even in a pretty small economy as well as remote consulting possibilities.

vericiab 17 days ago [-]

An option for entry-level remote work that might be worth looking into is working as an insurance claims adjuster. I work for a company that makes software for the insurance industry and from what I've heard from clients, it's not uncommon for their junior adjusters to be 18-year-olds that just graduated from high school. So it can be a true entry-level, learn on the job type of role and the pay is decent. It will probably be a bit harder to find positions that are both entry-level and remote but your familiarity with computers would definitely be a point in your favor. Other roles in the insurance industry (eg claims intake, insurance processor, underwriting support) might be an option as well. It won't lead to a career in tech but it may be easier to get into and can pay the bills while you continue studying.

For more tech focused options, you may be able to find remote internships or co-ops that don't require you to be a current university student. You'd be sacrificing some stability compared to a permanent position but they'll be expecting applicants to have little to no practical experience. A data analyst type role could also be an option, but I'm less familiar with the typical job requirements for that.

yakgwa 17 days ago [-]

Other people have recommended getting into support, and it really is a good option to consider.

A lot of the suggestions here are a bit more long term. There are entry level support jobs in tech that will give you valuable experience, `_now_`, for a pathway to engineering roles. Many companies will also give you a budget to buy courses, books, conference tickets, etc. Staying in support isn't a dead end, anymore, either. Remote companies will try to pay you based on location, but salaries can get very high once you stack up experience (120k+/year).

Regardless of future plans, you'll have an advantage over trying to learn while doing unrelated work.

Here are a few listings as an example:

<https://automattic.com/work-with-us/happiness-engineer/>

<https://kinsta.com/careers/>

<https://www.dropbox.com/jobs/listing/2603610>

<https://system76.com/careers>

If you want to talk more about it, lmk, and I'll email you.

throw_away30492 17 days ago [-]

I'm going to echo what exdsq said as well, support or QA roles would be a good in to a company, provide you with some tech associated experience even if its not directly coding. With QA you might be able to start out doing some exploratory testing, move more into scripting up some automated tests to gather more experience writing code even if would just be simple scripts to start. This will get you some experience with source control, working with other automated testers if the org has it and potentially lead you to pivoting to an SDET as you get more experience then based on your experience with that stuff you could pivot to development work.

Producing good work in your current position and having great communication skills can get you very far relatively fast in my personal experience.

Finding jobs through HN has worked for me in the past, in the search just type {month} {year} and look through the whos hiring threads ctrl-f for remote, look through the companies listings and see if they got anything new even if the post is a couple months old! HN can be a cheatcode to getting an interview rather than cold applying through monster.com or some shit like that.

Good luck!

rumblestrut 17 days ago [-]

Check out <https://www.launchcode.org> and see if you can get "in" going that route if they have one in your area.

Also, there's nothing stopping you from making things now, getting them online, and building on that experience. With not a lot of experience, I called up places and asked, "Do you need a website?" And got started that way.

Good luck!

AskHNremote2021 17 days ago [-]

I'll look into this. My thinking is that there weren't be anything for my area. But I am willing to check out anything at this point, because I'm on a time crunch. I'll pull up it after I follow up with those who have potential job leads. Thank you!

haskellandchill 17 days ago [-]

I would hire you. In a junior position. The software engineering job market is inefficient in ways that work in your favor and against it. Good luck.

rrherr 16 days ago [-]

Some remote jobs you could apply to right now:

Automattic Happiness Engineer: <https://automattic.com/work-with-us/#open-positions>

Zapier Support team: <https://zapier.com/jobs/#job-openings>

To help you get a remote software development job, consider doing a remote, part-time bootcamp with an ISA (Income Share Agreement). That means you pay nothing until you're hired. Some options include:

Lambda School: <https://lambdaschool.com/courses/full-stack-web-development>

Thinkful: <https://www.thinkful.com/bootcamp/web-development/topic?form...>

ChicagoDave 17 days ago [-]

Lots of good advice here. I'll add learn AWS. The video training on acloudguru.com is excellent. If you get a few certifications (udemy.com has excellent test prep) under your belt, that may open doors and it's all remote-enabled work. For a few hundred dollars and focused time, you can learn a lot about modern software development.

LoudPipes 15 days ago [-]

AWS has plenty of free AWS courses: <https://www.udemy.com/courses/search/?price=price-free&q=AWS...>

FreeCodeCamp on youtube has plenty of free, high-quality content.

crnkofe 16 days ago [-]

Here's my two cents. I'd strongly suggest as a junior hacker NOT to go job hunting without a decent portfolio (can be GitHub or similar) and at least one decent language or tech specialization. I've been recently through a lot of interviews as a senior and I can tell you the bar is very high. Making one or two mistakes on whiteboard coding is the end of that interview. Solving hackerrank challenges live while explaining how to do it is also quite hard and a very different skill from day-to-day software development. Also recruiters are trigger-happy in online interviews which are currently still the norm due to the pandemic.

I'd suggest to rummage through your existing network for opportunities. Also QA and IT Support roles with a bit of operations work might suit you since they involve a bit of software development but should be lighter on development skill-set requirements.

tmaly 16 days ago [-]

It seems like you already have some good tech skills here. Have you considered trying out upwork?

Hear me out, I was trying to figure out how to best go about sending out emails to people who sign up on my website. I try to listen to podcast episodes to learn some of these things. A well known copywriter from Australia was interviewed on the podcast episode I happened to listen to 2 days ago. His story of how he got started was brilliant. He went above and beyond with the projects he did on upwork. People that hired him were blown away by how much value he provided. He started raising his prices as word spread about the quality of his work. I think what he did is a very viable strategy for those getting started.

namank 17 days ago [-]

1. Google 5 help desk postings that meet your requirement 2. Make a resume with skills relevant to help desk 3. Take points from postings in 1 and add them to your resume in 2 3. Google a list of top job boards like indeed, LinkedIn and create a profile there 4. Google job boards specifically for help desk and make a profile there 5. Google companies that have postings for help desk 6. Everyday apply to 10 jobs on each of 3,4,5

While doing 6, apply to other jobs too like IT admin, customer service agent, bookkeeping, virtual assistant, secretary, exec assistant, and anything and everything else.

This is a game of numbers, apply to as many as possible as fast as possible. Keep doing this everyday, every week till you get it. Be creative, exhaustive, and bold at each step.

TurkishPoptart 17 days ago [-]

Very interesting. How did you find projects to contribute to in GitHub for documentation editing? I've always wanted to do this kind of work but haven't been able to find opportunities in it.

AskHNremote2021 17 days ago [-]

For GitHub, I looked at projects that interested me, but not necessarily having the skills to contribute code. So I would pour through the docs and find errors, typos, mistakes. These kinds of things. Cleaning up. Sometimes added documentation that was missing.

ilaksh 17 days ago [-]

I think that exact job of documentation editor must exist somewhere. But definitely technical writing jobs exist. A lot are not remote but I think you can always ask about that even if it doesn't say that. The pandemic may have made them more flexible.

<https://www.indeed.com/q-Technical-Writer-Editor-jobs.html>

<https://www.flexjobs.com/jobs/technical-writing>

<https://www.upwork.com/freelance-jobs/technical-editing/>

<https://www.fiverr.com/search/gigs?query=writing%2Bproof%2B...>

You can also contact agencies like Toptal and Robert Half.

If you want a different type of job then you need to have evidence or a convincing story that you have literally done that job before, either as an intern or through training/education or as projects on your own. But the fact that you have literally done proofreading and documentation editing in a technical context and there is a record on Github seems to be concrete proof that should be possible to leverage into getting paid work doing the same thing.

dfraser992 16 days ago [-]

Here is a repo of options for freelance technical writing:

<https://github.com/malgamves/CommunityWriterPrograms>

In the process of doing this type of work, you surely will learn new things and probably will help you figure out exactly what you might want to focus on / become an expert at / etc. You'll come to understand systems from a user's perspective which always good for developers.

c- 17 days ago [-]

As a HM if a candidate showed that on their github - it would be a yellow flag (nonsense changes for the sake of changes).

Just my \$0.02.

spike021 17 days ago [-]

As a software engineer, I'd love to see that someone is capable of putting effort into updating documentation, even with minor corrections and changes.

This is the kind of thing full time engineers typically don't enjoy doing, so docs stagnate and can quickly become out of date or inaccurate.

It's super useful if someone is used to a routine where they know how to dig into a project, find errors/inaccuracies, and work to improve them.

rement 17 days ago [-]

Fixing mistakes, typos, and errors are good things. Many articles and people suggest fixing typos in documentation for your first open source contributions. I would be interested if I found a new GitHub account with multiple accepted/pending PRs fixing things.

mgkimsal 17 days ago [-]

If they're presented as "I'm a code contributor" but it's solely non-code changes, then yes, it seems disingenuous at best.

But if they're accepted by the repo owner, they're probably not nonsense. And projects need better documentation. If the OP was applying for a junior 'tech' role of some sort, and demonstrated that experience of revising documentation to be up to date and improved... it's a step up from some people I've seen over the years (who leave behind bad/broken docs, for example).

worker767424 17 days ago [-]

Genuinely curious, I don't mean to be insulting, and a hiring manager will be asking the same question: you've been playing with tech for a long time and sound like you're interested enough to be competent. How is it that your work experience is generic, unskilled jobs? I realize it's a poor region, but it's still the US (no visa issues), remote wasn't unheard of pre-covid, and even poor regions often have something halfway resembling a DBA or sysadmin within an hour radius.

mrzool 17 days ago [-]

Finding the first job in tech, when you don't have any previous experience to show, is not easy. I was in the same boat for a long time. I assume what OP is trying to do is to find an entry-level job with the intention to move up from there.

Also, when you're broke, you often don't have time to look for a good position, because you need money right now, so you take up whatever job is on offer, which leaves you with no time/energy to look for something better, and you can't take the risk to just quit and follow your dreams because you don't have any safety net. You can easily spend years this way.

Of course, all this would not sound great in a job interview. That's why, when I interview people, I tend to focus more on the attitude, personality, and actual skills of a person, and avoid asking questions about past odd jobs, prolonged unemployment periods, and the like.

We all need to start somewhere. OP just happens to be starting now.

Good luck, OP.

leandot 16 days ago [-]

What about teaching English online or becoming a sales rep for a non-US company, being a native speaker is a big plus and there is plenty of business outside US. QA might also be a possibility with your resume - learn how to use Insomnia/Postman + Selenium + some more fancy tooling to impress, like <https://github.com/elixir-wallaby/wallaby> + being well organized should help.

jacob_rezi 17 days ago [-]

Hey you can use our software (<https://rezi.io>) to handle the resume. Just chat me and i'll upgrade your account.

zerkten 17 days ago [-]

As others have stated, coding development jobs are the best thing to aim for. They also give you a baseline that has value everywhere else. Others have provided good advice on this so I'll focus elsewhere.

Documentation writing is something that can get you in the door at interesting places. It can also give you time with technology in a less pressured environment to learn it and switch to more technical jobs.

Bigger companies have roles like this, but they are obscured because documentation writer is not a sexy title, and the role is more than writing. It can be part testing features or providing product feedback ("I tried writing about this feature but it's so complicated. Will people even use it?")

To work out some places to try applying I started with Microsoft. I'd ignore locations, even when listed because companies haven't updated their tools to reflect remote or are stupidly optimistic. Even if you got fired because you wouldn't work onsite at the end of 2021, the experience would give you a serious boost. How I found the roles:

1. Searched for "Microsoft documentation". Found docs.microsoft.com. 2. Searched for docs.microsoft.com on jobs.microsoft.com. 3. Found "content developer" roles listed. This is the job title stuff I was talking about. At the core it's tech writing. 4. Pivoted to searching "content developer" - <https://careers.microsoft.com/us/en/search-results?keywords=...>

Repeat that for other companies and you'll find different opportunities. There may also be contracting roles but it's harder to get into those initially. Writing docs for open source projects associated with these companies can help to make contacts. Red Hat is another place to consider for this reason.

It's important to have a portfolio of work. If you apply for a doc writer role you'd need to find examples that map onto what they produce. Also, look for opportunities to critique their work to some degree and be ready to suggest how you might improve their current work. Docs are becoming more interactive and interesting too, so any suggestions on how to make them more interesting will find more ears.

FloatArtifact 17 days ago [-]

I know what it's like to care for family full time due to disabilitys plus work. Hang in there! I hope you can find something that you enjoy and it pays the bills! Best of luck!

AskHNremote2021 17 days ago [-]

It's really tough, but this thread has lifted my mood. I hope I can find something, too! It seems there are a couple leads so far. I hope you are doing well. It's a tough position to be in, but we have to do it.

alfiedotwtf 17 days ago [-]

Thanks for writing this up. I think a lot of us reading this can sympathise as we were all in the same boat once (although your situation is harder given the remote-only needs).

I've found what's missing in our industry is a centralised place for new devs to get their foot in the door. It's a lot of hard work mixed with lot of luck, but we could do better... if you're looking to do YC S21 but don't have an idea, here's one with users waiting to bust through your door!

andi999 16 days ago [-]

Just to consider: for your first job in tech, you actually do not want to work remote (since the first 2 years you learn how things are really done in that industry, and learning from coworkers). Then you can do remote work (if you do not need to think anymore to do the work). I understand you have different constraint, but if you find a good tech position, is it possible for you all to move for like 2 years?

Layke1123 17 days ago [-]

Unless you love learning to code and love continuous pressure to learn even more, don't get an entry level as a software engineer. Try to find a job that only takes one or two deep level of knowledge in a tech stack, and keep that. What most companies want an entry level job today is someone willing to work 120 hours a week to meet arbitrary deadlines to maximize profits. Avoid FAANG.

Temporarily21 15 days ago [-]

What would be an example?

Layke1123 14 days ago [-]

Blue Origin. Don't do software there. They are just like FAANG but probably even worse when it comes to aerospace than Boeing or Northrop.

goat_whisperer 17 days ago [-]

Re: Eloquent JavaScript. I tried to read it when starting on my JS learning path, and I quickly grew frustrated.

It may be the perfect match for you -- but all I want to say is, if you get frustrated with it, you're not alone! Don't be afraid to try a different resource to get started.

However, I've revisited the book and I enjoy it a lot more after becoming more familiar with the language

andykx 17 days ago [-]

If I might give a potentially upsetting, but honest, opinion: you aren't ready for work if the content in Eloquent JavaScript is beyond your current scope of knowledge.

I am in the process of applying to entry-level jobs and I can assure you that I have not encountered a single technical interview that did not require knowledge beyond what FCC covers.

hypersundays 17 days ago [-]

If you do documentation editing well, you might try and look for a role as a technical writer or freelance as one.

Other than that, I would keep improving documentation on open source where you can and should. Then highlight that work as much as you can on your resume.

I wish you the best of luck and hope that you find your start!

pknerd 17 days ago [-]

I know you're not asking about but did you think about starting your youtube channel and share what y you know and learning? This helps in multiple folds, you get to learn things in depth as you're teaching. You get another Income stream and it could present you as an authority.

pabs3 17 days ago [-]

Since you mention Linux, you might be interested in paid internships in open source:

<https://github.com/fossjobs/fossjobs/wiki/Resources#internsh...>

costcopizza 16 days ago [-]

I also have a special-needs sister so I commend you 1000%.

Are you near any colleges or hospitals nearby? Those are probably your best bet for finding a local helpdesk. Don't forget YC's Work At A Startup and AngelList -- I see remote customer support posting there as well!

tehwebguy 17 days ago [-]

Lots of good advice about continuing to learn to code for the medium- to long-term.

For the extremely short term, consider looking for a listing of ecommerce shops (maybe a high traffic Shopify listing on a blog or ecom-adjacent site) and hitting them up about customer service roles.

deepstack 16 days ago [-]

If you are comfortable with command line, DevOp and system admin would be a good place to look. Since you are comfortable with mac(freebsd)/linux There is always need for cloud engineer, SRE. Did you also look at the startup company from Ycombinator.

sub7 16 days ago [-]

Look into Salesforce consulting. You can get trained for free or low \$ and make many times the minimum wage WFH. I know people making ~\$100/hr and they have more inbound than they can handle.

There's a big gap between demand and supply and it's only growing.

orasis 17 days ago [-]

I hire all my employees on upwork.com and Fiverr, but you will be competing against global wages.

ilrwbwrkhv 17 days ago [-]

I just got an email from this service: <https://developers.turing.com/> They hire specifically remote only. Check them out, since I am not looking for a job now.

alohaandmahalo 17 days ago [-]

Lots of great advice already given.

If you want a 1:1 advisor, I recommend Jordan Carroll: <https://www.linkedin.com/in/jordanscarroll/>

johnx123-up 17 days ago [-]

How much does it cost?

quangv 17 days ago [-]

Upwork? Be a freelancer, work for cheap to build up resume/portfolio. You can start raising your rates or parlay into a contract or salary position.

Upwork has been a place I use to get some gigs, some of my best clients actually.

Good luck!

AdamJacobMuller 17 days ago [-]

> Help desk work seems the most promising / practical

Send me a resume, adam@numsp.com

hiimtroymlure 17 days ago [-]

All I can say is good luck to you and hope you land something awesome!

AskHNremote2021 17 days ago [-]

Thank you so much.

prakis 16 days ago [-]

If you need technical advice or help learning programming languages like Javascript, ReactJS, Java, C#.. let me know. I can spare few hours.

korethr 17 days ago [-]

I agree with your assessment that a help desk or support position would be a good match for your skills as you've described them. You say you can't find any right now, but the jobs are out there; I know someone who just recently got hired for remote helpdesk work.

Don't constrain your search to looking for just tech industry helpdesks. Helpdesks and call centers can be found supporting all manner of industries and segments, be it remote education software, point-of-sale credit card readers, one of those medical machines that goes beep in hospital rooms, or those tablet-like computers a mechanic uses troubleshoot a car's

engine controller. Tech has permeated into society to a greater degree than even us techies sometimes appreciate, and wherever there's a broad enough deployment of something, there's going to be people who need or want technical help with that something.

Start looking up every company you can think of, big and small, new and old, and hunt down their job listings, looking for anything that sounds like it could be mainly tech support work, whether or not the listing actually calls it that. Also worth looking at are contracting and temp agencies, as often tech support is an out-sourced role.

Since you don't have a technical work history yet, look to the time you've spent tinkering, say, during your win98/Ubuntu dual-boot adventures, for example. Find times you had a hard problem to deal with. How did you solve the problem? What was your process? How did you deal with impediments to solving those problems? That's something you'll probably get asked in an interview for a tech support position, and it will be good to have some answers.

Do the same with your customer service experience. Come up with examples when you dealt with a particularly difficult or unhappy customer, and were able to help them solve their problem to their satisfaction.

Once you manage to land a tech position, become a sponge. Start soaking up technical knowledge. Seek to understand the Why behind the How. If you have specific tools, dig into those tools' nooks and crannies. See if there's ways to use the tools better. Become valuable as the guy who's really good at figuring out how to solve a particular class of common problem, or taming some arcane tool your team has to use. And if documentation is lacking, leverage and polish your communication skills to create or improve documentation.

Doing this will set you up for being able to move up from the basic entry level role, into something more advanced that hopefully pays better, whether that's moving up in your original company or finding a better position with another company.

pryelluw 17 days ago [-]

Co-organizer of PyATL here. I help folks like you land jobs. Shot me an email (in my profile) and I'll do my best to get You on your way.

mandown2308 17 days ago [-]

<https://remotejobb.com/>

upbeatlinux 17 days ago [-]

<https://remoteok.io/>

5600k 16 days ago [-]

If you're into Apple, why not apply to the Genius Bar?

bothersumman 17 days ago [-]

Look into Manage Care Organizations-insurance

908B64B197 17 days ago [-]

> I need to work remotely for family reasons. I have a special-needs sister and I look after my youngest brother. They are what's most important to me, which is why I don't want to relocate.

Have you considered they might benefit from moving to a new location as well? Better schools and services for instance.

AskHNremote2021 17 days ago [-]

That could be something for the future - but would require a lot besides money. I don't want to go too deeply in the matter, but relocation (at the moment) is not an option. I need a stable income first, and would need to arrange for them to be okay with the move as well.

hmwhy 17 days ago [-]

There are a lot of good advice given to you already and I just thought I'd share some of my personal experience with you since I some of your experience are similar to mine mostly because you appear to be self-taught.

I found out the hard way that "being able to learn fast" is a potentially very dangerous mindset to have when you are looking for an entry-level job for the following reasons:

- * You could easily open yourself up for exploitation by "sweatshops" (lack of mentorship, time to learn, and terrible pay) which hinders your career progression (assuming that you don't want to end up in one of those).
- * You are very likely overestimating your abilities for "better" jobs that have stricter requirements on non-code-writing skills, including computer science.
- * You can easily become a jack of all trades, master of none -- not in the sense that you are a generalist and a capable fullstack engineer, but in the sense that you just knows a little bit of everything and nothing well enough for an actual job. It's not that it can't be fixed and developed upon, but for your first job I personally think that's really not what you want, I hope you don't end up wasting time to get to that point.

freeCodeCamp is great to get started with (I have been through different version of the curriculum a few times for various reasons), but even finishing all of it won't get you even close to job-ready for jobs that have heavier requirements on non-code-writing skills.

That's not to say you won't be able to get a job after finishing freeCodeCamp -- with a bit of luck you may even be able to find a job with very good terms. However, if you want to compete for high-level jobs and/or want to continue to develop your skills properly even after getting a job, I highly recommend doing things that are more difficult than what you are doing immediately, and make that a habit.

Since you are reading Eloquent JavaScript (which probably means you care about more than just getting code written and things put on the screen), here are a few things that you might want to consider/get into immediately:

- * As many have already mentioned, solve challenges on LeetCode regularly and perhaps spend time read some of the forum posts. The reason is that there is little hint/guidance in the freeCodeCamp curriculum about what/how you could do for more efficient algorithms -- in fact, in many cases brute-force, naive, solutions would be accepted. You don't want to get into the habit of thinking that solving challenges in $O(n^2)$ time is okay when interviewers expect you to solve them in linear or better time complexity.

- * Eloquent JavaScript is a great book (I recommend adding You Don't Know JS to your reading list), but try to read it a few times at different stage of your learning/career -- you will likely learn something new/develop a much deeper understanding of abstract concepts every time.

- * Supplement your study with other courses such as those freely available on edX or MITOCW. If you are looking for a structured way to learn outside of freeCodeCamp, perhaps consider something like this: <https://github.com/P1xt/p1xt-guides> (not affiliated with p1xt in any way, I just came across her things when I first started learning a few years ago and find them super useful).

- * Get a personal portfolio up as soon as possible, *and make an effort to refine it to show that you are different* as you go - otherwise you will regret not having it ready when you need it. This is particularly important for people who are self-taught. Writing blog posts are okay (some companies seem to use it as a hiring metric?), but don't things that are just simply copying/paraphrasing/"distilling" without adding value -- it would just make you look very bad to those who know what they are talking about; talk about the problems you encountered, how you solved them, what you did to get personal projects done, etc. Even if it's not writing intended for interviewers/HR, you will still benefit a lot from writing things down (communicating technical details clearly is an important skill and definitely one of the metrics used by interviewers!).

I don't have much advice on entry-level remote jobs, at times it feels like they don't exist and, if they do, only in large organisations whose openings are usually extremely competitive. A few of my peers who were self-taught landed their first and remote jobs simply by reaching out and asking; I'm not sure about the details, but they're all hard-working people who evidently care about learning more than just writing code, perhaps it's not a bad idea to cold e-mail once you get to the point that you are comfortable with your skills.

I hope it helps! Good luck! :)

ktrulr 16 days ago [-]

Great advice. Could you share some advice/tips for an Individual contributor tech role for someone in 40's?. I haven't faced an interview for a long time. Familiar with latest technologies. I am not desperate for a job change. But still I think, one doesn't want face rejections since it will hurt. @OP. Sorry about posting something relevant to you. I hope you will find a job soon.

sibelius 17 days ago [-]

weworkremotely.com is a really great remote work resource!

chanux 16 days ago [-]

Also <https://remoteok.io>

wrapido 16 days ago [-]

Give UpWork a go

throwaway939333 17 days ago [-]

[flagged]

saagarjha 17 days ago [-]

> Prioritizing your family ensures that you will NEVER GAIN THE SKILLS TO WORK AT MICROSOFT.

Citation needed.

throwaway939333 17 days ago [-]

[flagged]

saagarjha 17 days ago [-]

A personal anecdote isn't enough to back the claim you made with the strength that you made it with. And, FWIW, personal attacks are not welcome on Hacker News.

giantandroids 17 days ago [-]

This really is unsolicited advice. You really cannot assume your situation and circumstance is identical to OPs. He may well love his family dearly and already feel resolute its right for him to care for them. To walk away from family (unless you have another valid reason, such as its a dysfunctional situation) can haunt for the rest of one's life.

Applications are open for YC Summer 2021

[Guidelines](#) | [FAQ](#) | [Lists](#) | [API](#) | [Security](#) | [Legal](#) | [Apply to YC](#) | [Contact](#)

Search: