 Rob K: Okay, thanks for sitting down with me today at Mamadou. The first question I want to ask you is, so could you give us a little more detail just on your engineering background? What kind of projects are, and obviously, there might be some things you can't talk about here, but what kind of projects were you working on at Gusto? What languages or frontend, backend code were you writing primarily?

  Mamadou: I

  Rob K:ed out in mobile. At first, I was in iOS for a few months, at least during internship. Then I returned and I worked a little bit of Android and I worked a little bit of full stack. But in recent times, I've been mostly full stack. I mostly react on the frontend with CSS for styling and Ruby on Rails on the backend.

  Rob K: And hey, one other thing I wanted, it's okay that we're recording here, but again, this video, none of this is going to be made public. I'm going to transcribe the video to a transcript. I'm just using the video to transcribe it. The video is not going to be posted or anything like that, like the audio itself, just so you know.

  Mamadou: No worries. Mostly full stack, mostly React and Ruby on Rails.

  Rob K: Okay. So most of the development work that I've done or looked at is all JavaScript, Node. It's all like, none of it's been on a mobile app. Can you just tell me a little bit about what are the main differences between the two?

  Rob K: I see. What's your favorite part of developing something in a browser or on a server compared to an actual mobile app or on a mobile device?

  Mamadou: I'd say building things natively. It's a lot different to the degree. Let me just turn this thing off real quick.

  Rob K: Sure.

  Mamadou: I mean, it has its own platform. When you're developing on web, the platform you're coding on essentially is you're writing into text files and you're loading things up in a browser. So it's Google Chrome and whatever you use, Visual Studio Code or Sublime Text or whatever text file you're writing in. For mobile, if you're doing Android, it's this platform called Android Studio. If it's iOS, it's called Xcode. You're essentially just potentially dragging and dropping elements and you're adding UI and you're plugging in UI and you're making it show certain data. But those are the major differences. It's completely different platforms. It also can be like different ecosystems, completely different stack overflow pages.

  Rob K: So if I was looking at a code base, and again, I'm a self-taught developer, so I'm kind of outside of the industry in a way right now. I mean, I'm trying to break into it. But if I'm looking at a code base in Visual Studio Code for, you said native, is that kind of the phrase I would use if I'm developing something on the browser?

  Mamadou: If you say that one more time. You used the word native, which I haven't heard.

  Mamadou: I see what you mean. So, native refers to like, so you're like, you can develop mobile apps two ways, right? You can develop them natively, which means you go inside Android Studio and you write Android code and you produce an Android app. Or you can write, go into Xcode, write iOS code and produce an iOS app. That means you produce them natively. What non-natively would mean is like, if you use something like React Native or use something like Flutter. What those are, I don't know if you're familiar with them, but those are like cross platform frameworks. So basically, you write your code once in React Native and then it produces out an iOS app and an Android app.

  Rob K: Okay. I'm going to have to look a little bit more into that. I'm not too familiar with any like, you know, I'm pure like JavaScript.

  Mamadou: What's nice is if you are Java, have you touched React at all by any chance?

  Mamadou: I have done like a few tutorials. I was waiting to kind of, I am like waiting to pick a framework. I think I should just make a choice here, but my thought was I'm just going to stay with JavaScript until I feel good enough in vanilla.

  Rob K: Just jump to React personally if I was you. There's no need to learn enough. Because React is in JavaScript anyway, so you'll learn JavaScript as you're learning the framework. And React is definitely by far the most popular framework. If you know that one, and a company uses a different one, they won't care.

  Mamadou: Okay.

  Rob K: But all that to say, I'm not sure if I'm If you know JavaScript and you learn React Native, you can just write, you can just pick up. It's the exact same thing you're doing to build a website. However, it's producing apps. You're using it to build apps. It's very easy to make that transition.

  Rob K: Okay. One thing I read or I saw, I did some interview prep here and it looks like you have a good amount of experience teaching code and working with younger students, middle school students, high school students, even some university level students. I want to get a little bit more into that, but I did want to ask one kind of how to improve as a developer question here. There's a little bit of preface to it. There's this question out here that all these interviewers, podcasters are asking. It's how do you improve as a developer or how do you get better at writing code? The standard answer they all give is, okay, you just need to spend time in your code editor writing code. Pick a project, spend three to five hours a day or however many hours you can put into it to get better at it. You're just going to improve. The question I want to ask is, okay, let's assume you're doing that to the best of your ability. Do you have any ideas on if you had 30 minutes to do something that isn't just that, okay, I'm going to spend three to five hours in my code editor? I'm going to spend 30 minutes, not a lot longer, doing this thing to improve my developing skill set. To give maybe one example of that, somebody might say, in addition to working on this project, spend 30 minutes doing leak code problems every day. Do you have any ideas or thoughts on ways to improve like that?

  Mamadou: Good question. I guess this is way longer than 30 minutes. I'll give two answers. The first one is this is definitely at 30 minutes. One thing I did a lot in high school that was really fun was I went to a lot of hackathons. Hackathons, at least when you're a kid, they're very fun. I don't know if you've ever heard the word hackathon, but essentially, you bring a bunch of developers and different folks together and you just hack at something, you build something in 24 hours, 48 hours, 72 hours. They're really fun when they're in person. Again, those to me, those will feel like, oh, I'm just sitting there and I'm coding and building a project. At least for me, I don't even like to code too much on my free time if I do it for a living. Hackathons feel more community-oriented. You can join something that is specific to a passion you have, whether that be music, whether that be save climate change, whatever cause, whatever thing it is. Hackathons happen everywhere, anywhere, all at once. You go to one, you meet some random cool people who are in different industries. It's also a cool way to network, right? Then within 48 hours, they're like, hey, you build this random thing and then you say, hey, these are my skills. They're like, okay, they work around your skills. I think that just that logic that fast-forward. I also like to work on software development. I think that's a fun way of getting better at coding without spending hours coding. That's one main way. Another way for me is to get better at understanding tech and what tech options are out there and what stacks and frameworks are out there. I do hear you when you say the native thing. Sometimes, it's very much like once you get past all the words and understand what frameworks are, you understand what all these things are. Then I feel like you feel a lot more confident in your skills to build anything. What that could look like is spending 30 minutes a day reading. I think that's as productive as coding, which is reading and learning about a company's tech stack. Like, how did they... Oh, wow, they're not using Ruby on Rails. They're using Java Spring. Oh, wait. How can you do backend on Java? Then you just read about it and understand the technologies available to you. Then you ultimately realize a lot of them just do the same thing. Yeah. It's something that I feel like over... I've been in my free time learning for maybe two years now, but really in the last four to five months, it's been daily. I'm doing that where I'm spending three to five hours a day building something. I do spend a lot of time trying to take in information from different sources. One thing I like about tech is that it seems like every day right now I can find a piece of information. I've heard the word native before, but I've never really looked at developing for mobile apps. There's just so much out there, but I do feel like I'm finally getting to a point where I just looked at these popular news sites and I'm going to take pieces and build this. It's a good feeling to be able to look at something and pull it together like that. It's a feeling I had in the last couple of days.

  Rob K: I saw on your LinkedIn that you worked with elementary school students, middle schoolers. That was a few years ago, right?

  Mamadou: That was back in high school. I was very fortunate. I saw on your portfolio page that you have an interest in empowering youth, bringing technology to groups that aren't typically using it.

  Rob K: When you're teaching an elementary school student, what's your thought process? Give me an idea of how you would go about teaching an elementary schooler a coding concept.

  Mamadou: I've been very fortunate to work for a bunch of companies that teach very young children coding concepts through different platforms. I came up with these on my own or anything like that, but I just got lessons and developed plans to teach certain concepts. The first thing is you just find something they like, a platform they like, and codify it. At some point, I was teaching… Apparently, you can code through Minecraft. I don't know if you've ever played Minecraft. I got to a point where you know how you can edit blocks and you can make blocks do certain things and you can make certain things appear. You have to get to the point where you're codifying it. We had many different lesson plans around that, which is, hey, add this code to here. That helps teach the coding concept while at the same time, these kids love Minecraft. It's going to teach them to get better at Minecraft. Obviously, everyone loves Minecraft. Something like that. You figure out an entry point and I think gamifying it is also key. The platform Scratch. Scratch MIT, a lot of it was designing lessons through there. Build certain games, build certain pathways, and allow creative folks to be creative, the creative students to be creative. It's a platform. I enjoy algorithms to solve most complex problems. Especially when you're using Minecraft or something, would you try to talk about some of those high level, I mean those are like abstract concepts. I think so though. I agree, yeah, they're very abstract. But I think, you may never

  Mamadou: up using the word DRY, but you will teach the concept, you'll teach an idea in Scratch that, hey, so you want to do that again, oh and then you want to do that again, and then you want to do it again. So guess what, if you just write again here, you can just run it as many times as you want. You could even indicate how many times you want to run it. I will say it's those mental concepts, because even now I actually still tutor, and I'm tutoring someone right now, and I even noticed, like literally, it's introducing someone to a for loop for the first time. It's a mindfuck. Like the idea that this I can be 0 and then be 1 and then be 2, but you could also make it 0, 1, 0, 2, 4, 6, and it could increment by certain numbers. It's definitely a mindfuck. But I think grasping those concepts early makes them much easier to understand later. You don't have the words DRY, but you learn those when you're in school or when you decide to study coding.

  Rob K: Well, one thing I've heard is the longer you are a developer, the harder it is to put yourself in in the place of like a junior where like i mean i was just learning about you know like i have that i had that mind fuck like a four loop like but but now as time passes and i write more four loops like you know it kind of fades away the the complexity of that concept but like i don't know if it's you know the complexity of the concept is changing or if it's just like i'm trying to think of the question i'm i'm asking here but like i mean this this student that you're tutoring like you're introducing them to four loops i mean you know like what's their you know what's what's their kind of response to like a four loop right now i mean you know what what are their problem points they're having if any i think i i think specifically is the fact that like there's this such a variable and they can change and then like understanding that stuff in detail and then also still having a zoom on be like wait okay but this code is still going to have build a calculator for me and then like you know but you still need to do all these nitty gritty details i'm trying to understand so i think this idea that data structures and how like you can say something is a set something is a map and like things can take different forms and but like they're also you're just writing them on paper but like you're not like you're writing them on paper but you have stuff written up all over the place so you have some kind of paper you're writing down things that we save some sort of data model and you have hundreds of other data models Geoff et al have that but like god Somehow, they represent these really abstract things and there's an infinite possibility, like infinite permutations of code you can write. I think it's also the daunting task of, when it comes to algorithms and data structures, there's literally an infinite solution most of the time. But the fact that you have to figure out and create one of them out of these codes, I think, these abstract things. I think one of the abstract things too is, it's not necessarily abstract, but one of the things folks have to learn also is your tolerance for failure. I think when you've been coding for a very long while, your tolerance for failure is so extremely high that as you're coding and it's not working, it doesn't mean anything to you. Right? You're still going to just keep adding things and debugging because for you, it's part of the process. But I think for someone who is new to coding and they're now coding and things aren't working, it oftentimes can feel like failure and feel like something's going wrong and I need to fix my code and X, Y, Z. But I think someone who's been coding for a while, you'd know that it's not that your code isn't broken, but this is how you code. This is just part of the development process. It's part of the race. But I guess to bring that back to teaching, I teach people failure is not even failure. It's just your code isn't working. That means you're coding. You're doing something right. Good job. Keep going. It's something I definitely try to tell myself. It's something that's out there a lot too. It makes sense, especially learning this stuff. Like there's... I love the idea of building a full stack app. I was making my first full stack app a month ago and I ran into this problem. Normally I'll run into a problem and I'll spend a session and by the

  Mamadou: of the session I'll be able to at least see the problem. There was definitely one point a few months ago where I was like, I don't know if I'm going to be able to solve this. I don't have a resource and I don't know what to do. Eventually we got there but it was one of those moments where I was like, I don't know what to do. I tried everything multiple times I thought to myself. I've been coding alone can be scary. I respect and commend you for doing it. I've been very fortunate where I haven't had to do it too much. I spent 20 minutes writing a Stack Overflow question the other day and I pasted in something from my code base. I got to the

  Mamadou: and I made it look like this is a great question. I hit post and it's like we've detected spam in your question. I'm like how is this possible? I spent 20 minutes writing this myself. I actually stopped. I didn't even post it. I'm like, I'm just going to sleep. I don't know how Stack Overflow can think I'm a robot right now.

  Rob K: Exactly. I'm the opposite of a robot. I know nothing.

  Mamadou: But it was something inside of that code base that tripped up something with Stack Overflow. Okay, so I do want to transition into kind of more something about layoffs here. I want to

  Rob K: with a broader question and that's, you know, in kind of preparation for this interview series, I sat down and I compiled a list of memos and like letters from these tech companies, like the CEOs wrote memos and letters to all of the employees of the companies. And they all kind of take the same format and it's like, hello, you know I'm deeply regretful to inform everyone that, you know, we have to make some changes today and one line that I saw kind of recurring in all of these was, and, you know, the line is, due to the changing like economic realities, we now face. And like it's a very corporate line and like I kind of want to just, you know, put that idea to you and ask you like how does, how do those memos and letters resonate with you like in those, those broad statements about like hey we as a company our economic reality is now different and by the way, just about most of these tech companies are still like highly profitable companies.

  Rob K: I see. What kind of software do you specialize in?

  Mamadou: I work mostly on backend systems and infrastructure. I work mostly on backend systems and infrastructure. I work mostly on backend systems and infrastructure. I work mostly on backend systems and infrastructure. I work mostly on backend systems and infrastructure. I work mostly on backend systems and infrastructure. I work mostly on backend systems and infrastructure. I've also been involved in the software industry and I work in the context of cloud service. I've since begun archiving and managing Database tools that make use of Microsoft Azure, Microsoft Office, Cloud every day, etc. That is my prioritization. I also base myself on Microsoft Office and AWS HUB.

  Mamadou: That's nice of you, John. Now, regarding the development of digital software, what do you think could lead to victory in the gaming industry? reading all of them in a row and I'm like, I mean it just sounds so strange to see like, you know, to see all these executives just like saying the same thing. They're all saying this line too, like, you know, I take full responsibility for this and it's like, I'm like, okay, like, I don't, what does that mean? Like, it doesn't mean anything when I read it. Exactly, yeah, exactly. And that, yeah, it's, I don't know, I'm glad, ultimately I will say I'm grateful it happened and I'm grateful I experienced this. Why? Because this is my first job out of college. I've never experienced a layoff. You hear about it, you hear people's parents talk about the financial crisis of 2008, all of that jazz, you know? But as like someone who graduated school, I never ever thought of a layoff, felt a layoff, so now that I felt one, you know, like, and I can't imagine what it's like to be someone who has a family, right? Someone who has a partner, someone who has someone to take care of and then just suddenly be laid off, right? I'm very fortunate that I don't have that. But if I was, you know, like, I would be very mad, you know? Especially if they're not expecting a piece that would make you more mad. It doesn't give you any time to prep. I haven't seen companies do that for a reason but still. I'm glad I've seen it and I'm glad I've experienced it.

  Rob K: Did Gusto put out a letter like that of Gusto's publicly?

  Mamadou: Oh, there is. No, no, it's on LinkedIn.

  Rob K: It's on LinkedIn? Okay. I've seen it on LinkedIn. Yeah, I mean, I have a feeling that any company that's put one out like that, it's out there. Yeah, it's definitely out there. Yeah, so I'll have to take a look at that, although it sounds like it sounds similar to a lot of these other ones. Yeah. But to get a little bit more specific, I want to, and this is kind of a sensitive question, but I wanted to get a more personal sense of what were the 24 hours like around this layoff? I mean,

  Rob K:ing with the day, how did it happen, I guess, is kind of the question. Yeah, I mean, the thing is, I guess it's very funny because I've seen the other layoffs happen, you know, and I guess I could talk about the big ones like Google, Facebook, you know, they just show up with a badge and the badge doesn't work, right? You hear those stories. And then to see it happen to you is always really funny because you don't expect it, you don't understand how mentally sudden it is. So I think about, I think the day before was a regular day, you know, I was coding, submitting, I actually had some code to submit in the morning, but I was like, you know, I'll just submit it in the morning. I even lined up some... Since Mina'sGenesis+, unless you've had their personal phone number. Like you legit cannot access them. They did not hit me. But even at this moment, I kind of still didn't know I was laid off because I didn't notice that I'd received some texts. But I had a lot of texts in the morning, you know, you're just waking up. I just thought it was weird. I don't know what's happening. Only until I Googled Gus DeLeon, and I saw the article that I was like, holy shit, I might've been laid off. But yeah, I think for me, it was just like, it was just confusion, anger, and like, it's the suddenness. You know, it's like, yeah, yeah. It's like if someone just like was there in the morning, and then they left in the morning, and you're like, what? What's going on? Yeah. Yeah. I think I went through a lot of emotions. I think I went through laughter, because I was like, this is so ridiculously funny. Like there's no way someone has audacity to do this. And then it became sad, and then it became angry, and then I found out another person I knew was also laid off. So we just kind of hung out and spent the day, and just did stuff, and that was really fun. But then towards the

  Mamadou: of it, it was quickly just trying to speak to folks, folks like lifting up my spirits. I think the worst part is you don't really know if it's performance based or not, because the way they're doing it. And like, at least for me personally, I do care about my performance. I do care about how I present myself, and I do care about how I present myself. Again, the keynote total, I will put the link to a course on how to get

  Rob K:ed in code with Like that is that can hurt someone a lot and I was very fortunate gonna have friends and folks to tell me like hey No, it wasn't performance-based So, I mean was there any indication? You know from management at all. I mean you said you woke up and no, it's quite the opposite like quite the opposite And I would even make sure I definitely says an anonymity but it's quite the opposite of the Of the reassurance that no, no, like things are fine. Like don't that's over there. That's not over here, you know Those are happening over there over here. So but quite the opposite also the other piece that makes it more frustrating, you know Yeah, you you very much don't expect that right, you know, especially especially if you're a naive New grad. Yeah, you know Right, I mean, yeah, there's a you know, I was in the logistics industry before I and one of the things that's kind of happens in that industry is you know when I put my notice in my my two-week notice I was very nervous that they were just gonna walk me out on the spot because the The you know the idea of I Mean there's just this, you know, Tabii Wow, Kayledchi transition, okay, awesome. Thank you so much for sharing. This is a heinous Garlic I feel thisFLU. I just first live leaving with my family. This is what happened I'm a like with my family you know 이걸 medios mean I help with training software that is run by people who work at CAD. You mentioned silos. Here's another problem with silos.

  Rob K: speaking to an other person. commission-based role, so like, I put my notice in in the middle of the month. If I would have gotten walked out, I lose my whole commission for that month, and it's like- Oh, for that month, wow, yeah. So I mean- That's a difficult thing. I was, you know, it's definitely, the corporate, like, it's crazy. I mean, like, you know, your bosses are telling, they're telling me, hey, we're family, like, you know, we're together, and it's like- Exactly, same, yeah. Like, the moment I have any sort of disconnect from the family, it's like, okay, now you actually just might be, like, severed completely, like- Yeah, no, that's very brutal. It's very brutal. I don't know what to think of that. I don't know, like, I, like, that's one question I do have. Is it an agreement that you accept as working in this corporation in, like, this capitalistic country? Like, is that just the unspoken agreement that, like, you know, you kind of fake it and fake family until they need to sever someone? Or is that an evil choice that someone, is that a bad decision that management is making? Like, I don't understand that completely. I mean, yeah, I'm with you. Like, I don't know either, and it's, from my

  Mamadou:, it's like, I don't know. have a lot of questions. If you How about now? Testing now? Can you hear me? I'm playing some music but I can't even hear my music. How about now? Can you remind me? No, I don't think it's you. It says my microphone is on. How about now? I can't hear you at all. Are you sure you're not muted? Okay, let me turn off my Bluetooth. I mean it looks like I got some... It looks like I have audio. Can you hear me? Yeah, I can hear you. Can you hear me? Okay, I can hear you now. I can hear you now. It might have been me. I just turned off my Bluetooth. So there's a chance maybe my AirPods just died or something.

  Rob K: I see. What was I saying there? Anyway, you kind of talked about it a little bit. My next question was going to be, have you been able to keep up with coworkers? It's like you're working with them every day. You're talking to them every day and then this day comes and are you talking to your coworkers still? Is there any connections there?

  Mamadou: Yes. Now I understand how amazing of an idea LinkedIn was. I never understood the value proposition of that. I was like, LinkedIn is useless for the longest. But no, LinkedIn is incredibly useful. You can literally just Google your coworkers. I'm at Google. Search them up on LinkedIn and add them as a friend and then reach out to them and say, hey. I've gotten that expression of sentiment. I reached out to some, some reached out to me and they're like, hey, I'm so sorry this happened. We totally miss you. I needed those validations to make me feel like, okay, I wasn't just a stranger on a team and they are equally mad.

  Mamadou: What has the process been since that day? You said it was emotionally jarring, which makes sense. Did you take a day? Did you take a week? How's it been the last couple of weeks?

  Mamadou: It's been two, three days I would say trying to understand it, wrap my head around it. I didn't tell nobody. I think it's been a good process. Last year, I ran a software department and I learned how to build reactive insights that contrast excellent with Tagen President and Sustainability not to just take advantage of our workforce and improve the workflow. I'm incredibly happy about the fact that my experience has contributed to development beyond only the framework.

  Rob K: So, when you

  Rob K: doing software, what kind of thing does gravitate towards science or engineering two-thirds?

  Mamadou: I think I just let myself be sad. I do like my favorite things, but also like I couldn't do all my favorite things right, because now I'm like thinking about money, you know? I'm like I like to eat, you know, and I can't even eat now. I can't go outside. It's too expensive. But I think after like three days, after I

  Rob K:ed telling folks, and I even tell like some some folks are a little older, my family, and I think they're essentially just like, hey, no, don't beat yourself up, you move on. I think the common sentiment I got was, there isn't anything to stress about. Like this is a norm, you figure out where to get back to, and you figure out how to move forward. And don't dwell on it, don't be sad, it's useless. It isn't, like there's nothing to, I don't think I was trying to rationalize it, rationalize the why, but I think everyone's told me, there's nothing to rationalize. Basically pick yourself up, move on, don't try to recognize it, rationalize it, just get on with your day. Not either choose rest, or the common sentiment advice I get is either choose, take a month off, hang out, chill out, or get back on the job hunt, brush up your resume, and apply for jobs. And of the two, I'm personally the type of person where I feel like I can't rest without knowing that I'm financially secure. So my logic is, I can't just do nothing. So I have to apply for jobs, and I just get back out there. My goal is to get something lined up, and just

  Rob K: it all over again.

  Rob K: I think I'm kind of similar in that way. It's one of the reasons I'm doing this interview series. I must have sent out close to over 100 applications to every major city in the U.S. that's east of the Mississippi.

  Mamadou: That's right. And like, you know, I mean, I just heard back from a position in New York today that, that, that didn't, you know, we're moving on without you. And like, that's a very normal thing to, to, to have happen, I think, for, you know, somebody without a background in tech to kind of like push into the industry. But I mean, yeah, it was like, one of the reasons I wanted to

  Rob K: this is like, you know, I, I want to actually build something that has more value than like the, the task tracker app or like, you know, just like the calc and like, I have done those, you know, building the calculator, like I've done those things and like, they're on my resume and like, you know, I'm, I'm doing all those checked boxes, but like, you know, I don't think, yeah, I mean, you know, I definitely wanted to try to build something that is, that could actually enter into like the main, I think that's a very smart move on that part. I think that's a very smart move. Cause I think you got to like build your own value, right? Like value isn't, value is you just interviewing people and adding articles to your website. And I think that's building value and value is this abstract thing that you just literally add on, right? Like, but it's probably the same website. You have the same skills before and after really, like you'll pick up.

  Rob K: That sounds incredible. Let's hear it.

  Mamadou: That is a large amount of work that I enjoy in all senses of the word.

  Rob K: Could you give us a little background on what Because it's one thing, I mean, you know, I

  Rob K:ed out with a resume and I talked to people in the tech industry and had them, you know, I iterated on my resume a couple times and I showed it to them, they're like, this is great. And they're like, you know, I worked on the portfolio and I showed it to some people and they're like, okay, this is, you know, these are the skills, like you have the skills is kind of what I'm hearing. But like, I mean, there is definitely something else that's missing. And I think, yeah, I think oftentimes that thing for folks that are just trying to learn how to code is a stamp of approval. Like that I think either comes with working at a company that someone can Google. Because again, it's one of those things where, you know, no one wants to take the risk. No one wants to, they're like, ah, you know, they worked on their own website, but that doesn't mean like they know how to like code, right. But that doesn't say anything. But I think it's the idea of company, it can be a school, it can be honestly these coding schools. I think people go there, they do learn how to code while they're there, but I think they're mostly there because that gives them the feeder pipelines that the company's like, oh no, this person went to General Assembly's coding school, went to Flatiron Coding School. I can say they know how to code then because we hired another engineer. I think it's finding what that's like. I think one of the best routes, though, just as a personal advice piece is also just hitting up

  Rob K:ups. Maybe the key is to work for a 5, 10 person company. Invest in them, add value to them, and when they go from 5, 10 people to 60, 80 people, they're becoming more legit and you've also worked there for two years. Those places are oftentimes more willing to take a chance on you. What you gain is a resume and gain experience and, you know, kind of remove the other stuff in your resume that doesn't have to do with coding. You know, you can layer it a bit there. And that gives you a lot more, a bigger chance of applying to the next spot, right? That doesn't, you know,

  Rob K: That's the other group of people I'm interviewing. Well, I mean, there's a couple, but like the two, you know, like I'm talking to people from companies who have been laid off, but like the other group I'm talking to is students from these coding bootcamps who are either just recently into a position or in the same position as me, you know, like where they're just part of the job search, I guess. Do you find that they feel similar to you in terms of like the, you need more, like the skills aren't enough? Yeah, I mean, I feel like, I think you kind of said it correctly in that, like, I mean, this, you know, you're right that it's companies don't want to take a risk and you can't blame them for like not wanting to just find a random person essentially in like pull them into an organization that is highly technical. From what I'm talking to them about, we have very similar technical skills, but they do feel like they've gotten a lot of value from that stamp of approval. It comes at a cost. Those coding bootcamps are like 50k. It's not like it's just a... It's a commitment, for sure.

  Rob K: Yeah, that's this country, more than anything. It's not even those bootcamps.

  Mamadou: Right. That's cheap for a school.

  Rob K: Exactly. That's very interesting. I also think there's a disconnect between HR and engineers and engineers interviewing. Say someone had the skills, I promise engineers would never care. The hiring manager, the engineering manager, he or she does not care whether this person was stamp of approved. Of course, there's value. I'm not denouncing the value of experience. You have experience and you've worked at these places, companies, you've seen problems. You've seen a lot of problems that you can fix. But I think oftentimes, HR and recruiters don't want to take a risk. They might not even understand what's on their resume. Of course, some do. I'm not trying to knock their technical prowess or anything like that. But they're just like, oh, I don't even want to put them to the next round. They have a certain level of power. And they're like, oh, I don't see an experience. Cool. But I think the thing about engineering and coding is there's no skill. People care more about skill. They care more just that you could get the thing done. I think that's a disconnect there. They don't want to take the risk on you, but engineers don't really care. I think an engineer is just like, I think if I interview you and you do phenomenal, you do better than the person who worked at Google or Facebook, I don't care. I'm like, oh, he knows React, he didn't know React as well or whatever the case is.

  Rob K: We're kind of at 1pm here. I wanted to

  Mamadou: on a question that is a little bit more, I guess, abstract here or just like a more general question. On LinkedIn, it said you were just, not just, you were a software engineer at Gusto.

  Mamadou: Yeah.

  Rob K: Are you looking at software engineering positions? Are you applying to any senior positions? What's that field looking like for you?

  Mamadou: I'm just applying software engineering. I was hoping to get something in the mid-senior position. Mostly searching, mostly being reached out to, but currently going through the process for sure.

  Rob K: What do you think, there's a lot of discussion about, okay, we have junior engineers and we have senior engineers. What do you think is a big thing that separates a junior engineer's code from a senior engineer's code?

  Mamadou: That's a good question. I've asked this everywhere. I've asked this a lot to senior engineers. My first answer is honestly purely confidence. Earther Cole faced confidence, so I think it's really cool—it's like being in an cartoon, but it's funny, your energy levels are higher ha ha ha ha. The person's first time is it's going to be, you know, looking around, you know, like they're struggling. But like, you know, then you get like that cool driver who's like doing his turns and going over. They're like, they both can drive amazingly, but like one is calm, cool, collected, under pressure. And I think that's what a senior engineer does. So that's one. And then the other thing, the difference between their code, because the irony is there is no difference between their code, right? Like the code will probably literally look the same because their code will probably do the same thing. But you know, senior engineers might be more modularized. It might be more folders that the code is in. The code might be shorter and more efficient. But those are all like child's play relative to like the thing being that like I might not get it. So yeah, I think their code will ultimately look the same. Also, I think another thing that is that is different is like the senior engineer is willing to take on those like, I think a good senior engineer protects their junior engineer, right? Like, like they will, they will pick out the projects that look that are sexy. I also buy dot-com between the industries and I'm in very tight Doing like crazy cool error handling infrastructure related things They were basically like holding down the foundation of the app that I could do all of this like pretty Work that such that the app will never break right and I think that's also the key difference. That's interesting Yeah, I mean it's to think like just to see a senior doing those like I mean That's what I would want to have in a my like a senior if I had it's like somebody who's like You know doing the air handling. I've tried to do some error handling on this like full stack Yeah, I'm like my god like To the Go like But okay I see you say let's see the engineers proposed work. They get to a point where they're like hey Actually, we need to bring in this framework. We need to propose they propose Hey, no, we need to spend some time on this technical piece and we need to spend this time as part of this code base You know as a junior engineer, just I'm adding things like Hey, yeah one quick question for you. Do you use what's your code editor that you're using I use Visual Studio code, okay? I use a program called iTerm. I don't know the second iTerm? I really recommend it. It's a tutorial to set up iTerm. I would just send you a Medium article. It's just a really powerful terminal. At first, I was stubborn. Right now, I don't really need it. A regular terminal is enough. But this one was way cooler. Thank you. It's awesome. Do you use Vim motions at all? I do not. I've never heard of them. Like, you know, like, have you ever heard of Vim, the code editor Vim? Yeah, the one that's by default on the terminal. Yeah, I mean, in like Vim motions are like instead of using your mouse to navigate the code editor, you're using like J and K to go up and down. You ever see anybody using this? I don't know. I've never. Okay. Just curious. It's one of the like, quick questions I wanted to ask, just get, get an idea of what like code editors are floating out there. I mean, you can go to like the stack overload survey and find that kind of stuff too. But you know, it's a good quick question. I'm gonna save this article. I appreciate this. No problem. But hey, I'm gonna stop the recording here and then just cover a quick couple things and we should be good, man. No problem.