[Transcript] AI Hustle: News on Open AI, ChatGPT, Midjourney, NVIDIA, Anthropic, Open Source LLMs / Modular CEO Chris Lattner: $100M Raise to Revolutionize AI Infrastructure for Developers

Welcome to the OpenAI podcast, the podcast that opens up the world of AI in a quick and concise manner.
Tune in daily to hear the latest news and breakthroughs in the rapidly evolving world of artificial intelligence.
If you've been following the podcast for a while, you'll know that over the last six months I've been working on a stealth AI startup.
Of the hundreds of projects I've covered, this is the one that I believe has the greatest potential.
So today I'm excited to announce AIBOX.
AIBOX is a no-code AI app building platform paired with the App Store for AI that lets you monetize your AI tools.
The platform lets you build apps by linking together AI models like chatGPT, mid-journey and 11Labs, eventually will integrate with software like Gmail, Trello and Salesforce so you can use AI to automate every function in your organization.
To get notified when we launch and be one of the first to build on the platform, you can join the wait list at AIBOX.AI, the link is in the show notes.
We are currently raising a seed round of funding.
If you're an investor that is focused on disruptive tech, I'd love to tell you more about the platform.
You can reach out to me at jaden at AIBOX.AI, I'll leave that email in the show notes.
Welcome to the AI Chat podcast.
I'm your host, Jayden Schaffer.
Today on the podcast, we have the pleasure of being joined by Chris Latner.
Chris is an incredible entrepreneur and currently is the CEO of Modular, a company that has raised over a hundred million dollars recently.
They just did a round of funding.
They're building some really incredible things in the AI space, so we're super excited to dive into all of that with Chris, talk about his journey and everything that they are doing over there.
So thanks so much for coming on the show today, Chris.
Well, thanks for having me.
It's great to talk to you.
Would you mind telling everyone a little bit about Modular and what it is that Modular does, like the problem that it's currently solving for customers?
Yeah.
Well, so Modular is a relatively new company.
We're less than two years old at this point.
We really decided to tackle the AI infrastructure problem in a very different way than most of the people are.
We decided to go all the way down to the hardware software boundary and tackle that low level how the hardware is driven, how the layers of software which make up these modern machine learning frameworks and the infrastructure and the serving components and all these things

got put together and rethink them.
And so we started from that layer because there's so many things in the space, right?
It seems like every quarter a new blog post comes out and they say, here are the top 57 things you need to know to do AI correctly, right?
Yeah.
And I don't know about you, but it's a little bit unsustainable, right?
We can't build and deploy systems where you need 57 different technologies all wired together.
So our philosophy is that you need fewer things that work better.
What we're trying to do is drive out a lot of the complexity from the system.
What that means is that we need to go and rethink things.
We need to redesign things from first principles.
And so Modular is taking a hard tech approach to this.
We're building compilers and runtimes and heterogeneous things.
We're working with the insides of TensorFlow and PyTorch.
We're trying to do this in a very user centric and customer centric way because nobody wants to rewrite their code, right?
And so what we're doing is we're building out that fundamental technology layer that we think can drive a lot more usability, a lot more programmability, accessibility, and things like this into this AI ecosystem.
Now, if you want, we can talk about all the components of that, but that's a high-level intro.
Yeah, yeah, I want to dive into that.
But first, I want to take a little step back.
So right now, you're the CEO running Modular.
This is an incredible company, solving a lot of problems.
What I would love to talk a little bit about your journey, what brought you to this point?
What got you into software?
What got you into AI to be in one?
Yeah, so I grew up in the 80s mostly as a nerd.
So I fell in love with computers very early on, first playing video games on the Commodore 64 and things like this.
Switched over to learning how to code in a serious way and worked my way up through things.
Eventually got a series of degrees in computer science and then entered the workforce.
Most of my DNA comes from Apple.
And so I spent 11 years at Apple building developer tools, ball sorts.
This includes low-level compiler technologies like this LLVM framework, but also Programming Language Swift, which I created, and then also was leading the entire developer tools ecosystem with Xcode and the entire developer platform.
Around the end of 2016, I fell in love with AI.
And so since that point, I've been on this hero's journey of trying to understand AI from all the different parts of this.
So I started in Applied AI at Tesla.
I worked on Google TPUs, scaling them, getting them to work, and really getting that novel

accelerator and needing new kinds of software to plug into TensorFlow and PyTorch and trying to understand that part of the problem.
Switched to the hardware side of things.
And so led a hardware team where we were building AI hardware itself and then building the software that goes with it.
And through this journey, kind of has been, have been looking at all the different pieces of this.
And through that, what it really reinforced is it's a mess, like whether you're on the hardware side trying to build a novel chip and you're trying to get people to use it because you need people to use your stuff or whether you're a software developer and you're looking to play something, this whole world is just so complicated.
This is kind of what led to Modular.
And my co-founder and I, Tim Davis is my co-founder, we got to know each other back at Google working on a bunch of cool projects together.
And what we realized is we need to kind of take a step back.
A lot of the people in the space are, you know, they're highly invested in technologies that were built five to six to seven to eight years ago, and they're all hill climbing those things, right?
Yeah.
But if you look at modern neural networks and modern AI, it's very different than it was back in 2015.
Yeah.
And so that's incredible.
Tell me, talk to me a little bit.
I just wanted to double, double click a little bit.
You said you, you created Swift, what, like, what, what was your role in, in Swift, working with Swift and, at Apple, obviously very incredibly popular, the programming language that is used to create all iOS apps and everything in their app store.
So talk about that.
Yeah.
So I joined Apple in 2005 and one of the projects I was able to work on there is this LVM framework that I started as part of my, my graduate research.
And I was able to bring LVM to Apple, productize it, and then went on kind of this journey of replacing GCC with an Apple.
And so by the time we got to 2010, that's when we'd launched Clang, which is a C and C++ compiler.
And so that was a full stack replacement for GCC, for all of C, C++ and Objective C, which is the technologies that Apple uses.
Now coming out of that, C++ is, how should I say it?
It's a wonderful and very rich language, has many powerful features.
Implementing it is a very fun and interesting technical challenge.
Having launched that.

An optimistic spin on it.
Yeah.
I'm being nice.
It's, it's, it's perhaps not beautiful, but it's very pragmatic, right?
There's a lot of great and many layers of technology built into C++.
Coming out of that, it was kind of the motive saying like, okay, well, prove that we could do this, but really is C++ the thing?
And so at the time I was managing a large team of quite, quite a few people.
And so it's just nights and weekends, hobby projects started and said, okay, well, let's build, build something next, right?
Let's see if we can get the iOS and the Apple ecosystem to move from Objective C onto something new.
And so I spent probably a year and a half just in my spare time hacking on the thing and building what became Swift, got to the point where a year and a half in it was interesting.
I kind of understood it.
I'm the kind of person that needs to build something to really understand the thing.
And then I approached management at Apple and said, hey, I've been working on the thing.
What do y'all think about it?
And they're like, oh, well, that's interesting, sounds complicated, but I had a good reputation.
And so they're like, okay, yeah, you can have one or two other people work on this.
And so the project slowly grew within Apple.
By the time we got to 2014, we launched it publicly.
And so we said, hey, app developers out there, you can build Swift applications and submit them to the store.
And so that whole journey was epic, went on for multiple years later to open source it later and do a bunch of other things, continue to be involved in the Swift project even after I left Apple.
And so, yeah, Swift is a really, really great project.
And I think it's really been widely adopted within the Apple ecosystem.
And Apple's using it for everything from writing kernel extensions to writing applications for UI and things like this, which is really, really quite wonderful.
This is incredible.
And I'm sure you have many lessons and takeaways that you get from this experience.
For me, it's so cool to see someone that, this was your hobby project within the company, right?
This wasn't your job.
I didn't say, hey, Chris, come in, build Swift with like a hundred other people.
Like this is obviously a very, a very important piece of software.
So you would imagine that like it could have a huge team.
This is your hobby project.
You're an entrepreneur, right?
You're starting your own thing inside.

And then it ends up scaling and being this huge thing that's a major part of the company.
And for its worth it, that's not unusual for me, right?
So LVM itself started that way.
So LVM is this compiler technology that underlies Clang and Swift and Rust and Julia and like all TensorFlow and PyTorch and like all the stuff that needs high performance.
That started again as a hobby project.
So working with my advisor back at university of Illinois, he was having me work on some stuff and he's like, okay, well, how about I like create something more interesting to nerd snipe him into allowing me to work on this thing.
And so LVM started from that.
Years later at Apple, right, started working on getting LVM to be production.
Had a big team built around it and a lot of really good people working on this and said, okay, well, everybody knows it's impossible to write a C and C++ parser, right?
What does impossible mean?
Like is it impossible or is it just really hard?
And so again, worked on what became Clang for, I don't remember, it feels like a year or so, something like that, just kind of understand it and kind of again, got to the point where it could hit some interesting metrics, it could prove the use case was a long ways away from being completed.
And again, brought to Apple management and said, hey, wow, would this be cool?
And it was super funny.
I remember my manager at the time is like, oh, that's super weird.
Like I didn't think that you cared about languages.
I thought that you just were a code generation person.
I'm like, well, I like working on really interesting technical problems and learning cool things.
And so, and so, you know, I've kind of have this habit of wanting to build things to understand them and then work through that and bring it to production and all phases of that journey are super exciting and interesting to me.
That's amazing.
Obviously, that's how you can build crazy and impressive things that you're passionate about.
So that's incredible.
One question I will ask you is starting from this sort of like you're creating these languages you're working in this space, what kind of brought you into AI and what you're currently doing today?
Like was it that you were intrinsically interested in AI or is it just kind of like the natural progression of where your industry was going?
What brought you into that space?
Well, so I really fell in love with AI back in 2016, right?
And to me, it started from just very much a, this is a new kind of science time.
Okay.
2016 was a long time before chat GPT, right?

And so this is where, you know, it was a combination of what people were doing back back in 2016.
It was just recognizing dogs and cats and images and things like this, right?
And the thing to me that was fascinating is that, you know, there's no good way to do that with traditional software.
Like you can't write a series of for loops and if statements to find out if something's a cat versus a dog.
It doesn't really work, right?
But deep learning was like a new form of programming.
And so as a person who cares about developers and of like being able to build cool and innovative software is like, oh, wow, there's a thing here, right?
And so from that point, then got involved in the system side of it.
And so the compilers and the big accelerators and stuff like that.
And what I realized is that, okay, well, not only is it an algorithms question, it's also hardware, it's also new kinds of compilers, it's this new generation of technology.
And what a modern machine learning framework and system does is it produces a higher level of abstraction, like the graph, for example, right?
And one of the things that's really, again, I fell in love with is that it's a fundamentally different kind of programming.
And it has fundamentally different kinds of powers than what traditional programming language is enabled.
And so, you know, I remind people like, again, 2016, that's when people were talking about let's go democratize AI, right?
And you know, you fast forward to now and we have people that are working in Jupyter notebooks writing Python code, and they're spinning up clusters with exoflops of compute, right?
So something happened there, right, and my journey through Google is helping build a lot of that technology.
And there's something fundamental there.
And so one of the challenges with today's systems, though, is that a lot of the today's systems end up being these point solutions that are really optimized for one specific use case, one specific piece of hardware, one specific research group, one specific kind of model, you know, whatever it is.
And so a lot of what modular is doing is saying, okay, let's learn from that.
Let's take all the learnings that the industry has done, let's take all the research, let's take all the many cool point solutions and systems have been built, let's build a world of class implementation of that technology in a really easy to use, actually good system.
And so again, what we're doing is we're saying, okay, well, we don't need to do research here.
We have, you know, modular is a company based on six or seven years worth of research from both Tim and I and many of the people in our team, we've been building all these systems.
And so getting the chance to do it again and do it right is really quite powerful.
That's amazing.
Incredible company, talk to me a little bit about, so you're like, like the founding of modular, how this thing got started, what that looked like, what was the, like paint

me a picture here, right?
You're working in these corporate jobs, you're at Tesla, you're at Google, you're at Apple and you're doing these kind of things in your career.
Like what was the moment when you were like, oh my gosh, like I need to start my own company, I need to co-found this with Tim, like what was the conversation?
Did you like, hey, Tim, like I have this idea or Tim call you up, like what did this look like?
Yeah, yeah.
So, so time frame is a little bit more complicated than that actually.
So I left Google in January of 2020.
Okay.
That was right before the pandemic struck.
And there's a whole story there, but I left Google because I really want to be working on this ML infrastructure.
And at the time, I kind of had this feeling that PyTorch was winning and TensorFlow wasn't and it wasn't the right place to be, but I also wanted to go work in a startup.
And so the smallest company I'd worked at at that point was Tesla, which is not exactly a tiny little company, right?
And so I joined a hardware startup.
And so that was super interesting because again, I got to work on the hardware software boundary, but this time seeing how the chips were made.
And so learned a tremendous amount and worked through that, the pandemic struck that led to lots of interesting excitement company, built up new kinds of technologies, learned how all this stuff was built and then understood at a much deeper level how accelerators had to fundamentally work and therefore what the software had to look like.
It's like, because understanding both the physics that the hardware people are dealing with when putting putting transistors down, but then understanding the abstractions you can build on top of it together was something that was really eye-opening to me and really helped complete my worldview.
At the same time, Tim had stayed at Google and so he he he rose into the ranks and was on the product side and was part managing all the mobile stuff, the data center stuff, like all this infrastructure for AI there.
And we kind of for both reasons, for both our parts of the journey kind of got into this point of where we realized nobody is really working on doing the fundamentals, right?
People have built all these systems and they're again, they're hill climbing them, right?
The PyTorch team is going to make it more PyTorch, the TensorFlow team is going to make it more TensorFlow, the NVIDIA team is going to invest in their stack, the Apple people are going to invest in their stack, whatever, right?
And there's nobody that's looking to unify this tech, right?
Of course, Meta's not going to work on TensorFlow, right?
Everybody knows that, right?
Yeah, yeah.

And so we got this idea to say, okay, well, let's actually set up a company that can be independent.
And being independent is actually really important because AI is so big, it's so important to so many different people that roughly everybody's competing with each other, right?
All the cloud companies want to kill each other, all the hardware people don't get along, all the LLM companies don't get along, right?
Yeah.
The TensorFlow, the PyTorch people, they're like dogs and cats.
They're both delightful, but they don't get along super well, right?
And so what we decided to do is say, okay, well, let's be an independent company and let's be focused on making the world's best thing.
And what that means is, let's be real, some people use PyTorch, some people use TensorFlow, right?
Particularly if you're talking to enterprises, they've been building and deploying AI into their systems for five-plus years in many cases.
And so they have a little bit of this, a little bit of that, nobody really knows how it all works anymore because the people who built it maybe have moved on.
And so there's this big, big mess.
And so what we decided to do is say, cool, let's clean up this mess, let's build this unifying technology that can work with both TensorFlow and PyTorch.
And that's production quality, it's commercially supported, it has really phenomenal performance.
I mean, we can often improve performance by like two to three times, for example.
Wow, impressive.
That then enables people to have bigger models, which is also a really big deal.
And as you kind of lean into this, you start to realize, well, this stuff is so important that the fact that the complexity is fractal and it's just exploded means that actually nobody, it doesn't fit in anybody's head.
And so there's this huge gap between what theoretically can be done if you have a team of 100 people that are the experts in everything versus what people are actually doing.
What we're trying, in practice, and what we're trying to do is we're trying to close that gap between theory and reality.
Okay, very impressive.
And so, yeah, just bringing it back to the beginning of modular for a second.
So you guys have this vision, you see you need to connect these technologies, PyTorch and TensorFlow and all this stuff.
So what was the conversation that you had initially with your co-founder?
I mean, you guys had worked together, but was it like, hey, let's do this thing, let's put this together?
He was working at Google, I'm assuming, so we had to quit and come in?
We both had full-time jobs.
So he and I knew each other, we had become friends.
But as you say, it's never that simple, right?

You have to be, your timing has to be right, you're both giving up the opportunity cost of doing something else.
And so we both could take a different big company job and be the lead of whatever, of AI at some other really established important company.
And so we had to make sure that we aligned on what kind of company we wanted to build, how we wanted to approach that.
One of the things that was really important to both of us is to start with the business model.
Yeah.
In 2021, that was kind of a novel concept, but it turns out it's a pretty important...
A little bit more popular today.
It seems like a good thing.
I'm a little bit old school, I kind of move back to the, try to build core value and solve problems for people, and particularly if you alleviate suffering, people are often willing to pay for that.
So yeah.
So yeah, so we worked through all of that and then decided to incorporate the company.
And so we started in January of 2022.
That's incredible.
And I mean, you guys have obviously seen a lot of success from the very beginning.
Again, like I mentioned at the beginning of the podcast, a big congratulations on raising a hundred million dollars.
You've just raised recently for modular.
Can you tell us a little bit about how this is going to accelerate modular's mission to fix AI infrastructure for the world, what this is going towards?
Yeah, totally.
You've covered us before, so I'll repeat your own words.
We're using it to build a team.
So one of the things that's really, I think underappreciated is that the technology that we're building is really, really hard and requires a lot of really specialized experience.
And it's actually perhaps more difficult to hire a compiler engineer than an engineer who knows how to train a machine learning model, right?
Just because in terms of talent, people have been training neural nets and learning how all the AI application layer works for many years, and many computer science graduates come out of school knowing this.
But the technology stack, the hardware-software boundary is really niche.
And so what we're doing is we're competing with, effectively, the big companies to try to hire the world's best talent and build a really unique team.
And so we've been doing that quite successfully.
We have some of the people that have built TensorFlow and PyTorch and Onyx Runtime and TVM and XLA and all of these systems that people have been using.
And so we're pulling together these folks and giving them the opportunity to build

something that's really fundamental and profound, where before many of them and us
were all kind of stuck, hill climbing, right?
And stuck in the constraints of the host organization.
Now we can actually build the right thing for the world.
And so the primary spend is on people.
It turns out people are very expensive these days.
And one of the things that's very unusual is we're not just spending on,
you know, AI compute hardware or something like that.
OK, very interesting.
Yeah, and I think this is an incredible, incredible use case.
Building out that team today is such an incredible, you know, value add for the company.
Something I'm really interested in talking about that a lot of people have asked about
in relation to your company is Mojo, right?
So you as we talked about at the beginning of this, which kind of I was interested,
you've had a really fundamental role in developing Swift over at Apple.
So Mojo is kind of touted for those that don't know as like a new programming
language for all AI developers that combines the usability of Python with the performance of C.
So what was the kind of idea?
What was the directive behind creating Mojo?
How does it differentiate from existing languages like Python or Julia?
Yeah, yeah, great question.
So let me give you, let me tell you where it came from first, and then we'll tell you
where what it accidentally means to the world.
So when we started in modular, like we were talking about, we wanted to solve
some very fundamental hardware, software boundary kinds of problems, right?
And so how do you make a GPU go burr?
How do you have a massive cluster of machines talking to each other?
How do you build this layered software architecture that is usable by Python
and by millions of machine learning engineers worldwide?
Right. And so the state of the art had been to write everything in CUDA,
write everything in C++, write everything in Python and have this unholy mix
of all these systems that don't really talk to each other very well.
But kind of with enough duct tape and bailing wire, you can make this thing work.
And so as we're tackling this, there's a bunch of other really fundamental
technologies like kernel fusion.
And there's there's a whole bunch of tech that you need to be able to program
these heterogeneous machines, because it's not all about a GPU.
Turns out that GPUs don't do data loading from spinning disks.
You need a CPU for that. Right.
And so as you're actually building out an AI technology stack,
you get exposed to all this complexity.
And so what we did was we said, OK, let's go and build a really novel compiler.

And we started without any syntax at all.
So we just handwrite the internal representation that the compiler used
and proved that we could generate state of the art performance
and we could express the things we needed to do.
And we built out this core technology and we got to the point where we said, OK,
it's seeming to work, but now nobody can use it
because there's no syntax for it.
And so what we did was we went shopping and said,
what syntax do we think we should use for this?
And so, you know, you could either do like Swifted,
which is you invent a new programming language.
We could use C++.
We could use an existing thing off the shelf.
But we realized that none of them would work very well.
And so what we decided to do is, you know, for a person and for a team
who's built programming languages before, you say, OK, well,
why don't we just do something new?
But instead of inventing new syntax, why don't we just follow Python?
OK. And the reason for that is that everybody in AI already knows Python.
Right. And like most people love it, except for performance
and certain limitations and challenges, right?
Most people love it.
And so if we can take the part they love and then we can solve the problems with it,
we think we can have something really, really nice.
And so what Mojo is, is it's a completely new compiler runtime stack.
It's super efficient, designed.
It's not designed to be an improvement to Python.
It's designed to work backwards from what the physics allows us to do with the hardware.
OK. But then it's a member of the Python family.
And as it builds out, it becomes a full superset.
And so what you can think of it is it becomes to a programmer like Python plus plus,
where you can you can take forward everything you know about Python
and you don't have to retrain like everything you know will still work.
But then if you want to add types to it, if you want to adopt new features,
then you can get better performance.
You can deploy with fewer dependencies.
You can get a lot of the benefits that Mojo provides you.
Ah, OK. Yeah, I see.
I see a lot of really incredible benefits in that and what you guys are building.
I think it's probably would be a very it probably seemed like a very daunting task
for most companies and people to to create something like that, of course,
with your experience and and like also just kind of hearing your background

and some projects you've worked on.
You seem like the kind of guy that if there's something daunting,
you just, you know, go ahead and do it anyways and jump right in.
So well, yeah, I mean, I think there's an aspect of that, but it's also
let's not get ourselves building a new programming language
and doing it the right way is a huge amount of work.
Yeah, seriously.
This is a huge amount of work.
And it's like we're talking about before, extremely expensive to do this.
Right. And so we're not taking the short path to have a quick hack to make a demo.
We're taking a much more difficult path knowing that it's difficult.
So full, fully informed, have been there, done that.
But believing that the outcome will be worth it.
Right. And so that's that's one of the things that I believe in where,
you know, I believe in AI, I'm an maximalist.
I think it can reinvent so many experiences we have
and that there's so much technology here.
And so I'd like to do is I would like to make it much more accessible
to people so that people can not just train the model
that they can actually deploy it, right?
So they can invent new algorithms that fully utilize the hardware.
Like these things, you know, I I just believe will lead to progress.
And so the fact that you have to do really hard stuff.
I mean, to your point, isn't daunting because I like to do that kind of stuff.
But but to me, that that's just OK.
Well, is it worth it or not?
Yeah, is the investment in time and energy and things like this
going to lead to something fundamental?
And I believe the answer is yes.
Yeah, that's incredible.
I think having that conviction obviously is been a huge asset to you.
This is why you've gotten so many incredible things done.
I think modular has been such a success.
One thing I'm interested about with modular,
I'm wondering if you can elaborate a little bit
on how the modular compute platform dynamically partitions
and distributes models with right billions of parameters,
what kind of efficiencies and scale can developers expect?
Yeah. Yeah.
So so my background, as we're talking about,
goes back through decades of working on compilers.
And so I don't consider myself to be just a compiler person, air quotes.

But I know I know when and where compiler technology
can be particularly valuable and how to use it the right way.
You know, use it for great justice, not for not for full employment
of compiler engineers, right?
OK.
And so when I saw the rise of these AI systems,
so you take TensorFlow, for example,
the core the core abstraction in TensorFlow is a graph.
And a lot of the way that TensorFlow is built initially
was by kind of distributed systems people.
And so a lot of the initial architecture of TensorFlow
was to scale this graph across a cluster.
And if you have a graph, you can have each node turn into message
sends and things like this.
And so when a compiler person or a person with background
in compilers like me comes into this, I start to realize, OK,
well, this is actually a much more fundamental concept
than just something that, you know, just a machine learning thing.
And so bringing that technology, the compiler technology,
the distributed compute, the low level runtime,
the code generation, like all these pieces to bear,
that allows you to really understand what the graph is doing
in a way that people haven't been able to do in the same way before.
And so there's a whole bunch of this technology
that then allows you to, for example,
partition a model across multiple machines
or do it in parallel horizontally across thousands of chips.
Right. And so and so it really comes down to using a graph
and understanding that compute declaratively
instead of looking at for loops and semicolons and things like this.
And that's one of the reasons that Mojo and the AI engine
that we're building is an epoch of technology
different than something like Swift or Rust or C++.
It's just like a different category of tech, of course,
in the simple case, when you're running on one CPU or something like that,
it's the same, but allowing you to scale out
to this much more complicated heterogeneous computer is pretty cool.
I think quite fundamental.
Super cool. Yeah. And it does sound very fundamental.
I'm wondering, you know, based off of your ability
to do some really incredible, incredible tech, you're attaching,
you know, you're allowing people to work with TensorFlow, PyTorch,

all these different things.
I wonder if you could share some, you know, real world case studies
or success stories from people currently using your tech
and seeing some of the benefits from it.
Yeah. So, I mean, it's really hard to give you one answer
because we have thousands of people, hundreds of thousands of developers.
I mean, just today, somebody kind of...
So as we're recording this three days ago, we just launched Mojo as an SDK.
So you can actually download Mojo and use it locally on your Linux box.
Oh, very cool.
Already, people are like rewriting Lama models in Mojo and they're like
doing incredible things.
But the thing that's really cool about AI is it spans everything.
So we're working with internet companies doing
recommender models for, you know, movie prediction,
like maybe you should watch this movie next.
We're working with people that are doing like fluid dynamics
and things that aren't really AI, but are using AI tech,
working with people that are in the generative space.
And so there's, I mean, AI goes everywhere.
Cars have a ton of autonomous systems in them.
And so it's this is one of the fun things about AI is it's not.
It's not a cat detector, right?
It's a fundamental new kind of technology that gets used in pretty much
any place that you want to interact with humans or with the natural world.
And I look at it as, yeah, it's the best way to deal with the modality of real life.
Yeah. Yeah.
Very cool.
So something that I would be very curious to hear a little bit more about.
I know that Modular's cloud compute platform is set to launch in 2024.
I'm wondering if you can give us a little bit of, if you talk a little bit
about some exciting things we can expect from that, you know, you know,
training at scale and managed and BYOC, all that kind of stuff.
What are some things that people can expect with this launch?
Yeah. So what we've done is we've we've we've approached the market
with with the goal of meeting customers where they are.
OK. And so a lot of people are using a lot of technology that's
as we were talking about, super fragmented and very complicated.
But they've built a lot of existing systems around them.
Right. And so our approach and our today today products
are really designed to meet people where they are, even if it's unfortunate.
OK. And so and so our first product isn't the AI engine.

It works. It's kind of like a Docker container.
So the cool thing about this is you can run this on your cloud, any of the cloud.
You can run this on prem.
You can run this laptop. You can run it at any system you want.
You can orchestrate it. You can do whatever it is that you're doing.
If you have crazy security stuff, it fits and it just works.
Now, the challenge with that is that you have to build all that complicated,
crazy stuff and people are asking us for more, right?
They want to defragment all of that stuff built on top.
I mean, so many enterprises say that they want a unified platform for all their teams.
And so what we're doing is we're starting from, again, we're starting at the bottom,
build, build systems that drop and replace and embrace people where they are,
but then give them new things.
And so our our hosted platform becomes much more like a SaaS product
where we'll host a compute for you.
And and therefore take a lot of the burden of scale, maintenance,
reliability, deployability, all this kind of stuff off your plate.
So you can focus on building your applications on top of AI instead of
having to babysit it and, you know, decide which ML ops platform of the day
is the right one for you.
Right. Right. Yeah.
And I'm sure that that saves people a ton of time.
And really, the more headaches like you kind of mentioned, the more pain you
take away from people, the happier they are, the more they love your platform.
That's really exciting, a very, very cool and excited for the launch of that next year.
So, you know, as we kind of wrap up this this interview,
I would be really curious to ask you, you and your team have worked on
a lot of really incredible technology, you know, PyTorch and all sorts of
different things that are really integral to AI today.
And I'm wondering if you have a piece of advice for developers,
for companies that are currently implementing AI into their organizations
from like from a high level perspective, what's a piece of advice
or you could give to those people?
Yeah. I mean, to me, the thing about AI is it's so exciting because it is.
I mean, even when you go back a year ago, people were kind of skeptical.
They had heard about these things.
Some people were thinking about, do they need an AI strategy?
But now it's very clear that AI is here to transform a lot of
product experiences, a lot of the expectations that people have.
And so it feels sometimes like it's difficult to keep on top of things.
Right. And I guess one of the things I would just encourage is
unless you're a researcher inventing new algorithms,

yeah, just assume that new research is going to come out.
OK, I assume there will be new models.
Assume there's going to be new innovation.
Assume there's going to be new hardware.
And so look to a way so you don't get stuck on one thing.
Right. And so you need a little bit of insulation so that you can benefit
from new research, you can benefit from like this.
This world is changing so quickly.
Right. It really is important to not get stuck.
Right. And so that's one of the things where, you know, it's hard,
but this is where, you know, we're trying to help level up the world
and make it so that people can actually work with all the technology,
even if it's changing because the research innovations are just amazing.
Totally, totally.
I love that advice.
And I think, you know, that is that's so crucial.
And I guess that's really the the massive value of modular.
Right. Because all these companies would be stuck with all this tech debt
or like essentially the it's hard for them to move with the times
and come with everything new that is coming down the pipe
if they're not able to integrate with it with their old tech.
So modular kind of helps bridge that gap and does some impressive things.
Thank you so much, Chris, for coming on the show today,
for sharing your insights and your story.
Really incredible, really inspiring for a lot of people.
Love what you're building.
If people are interested in trying out modular as some of your platforms and tools,
what's the best way for them to find that and, you know,
kind of look at some of the things you're building?
Yeah, you can go to modular.com and there's get started button.
You can download it and start using it today.
It's incredible. Awesome.
And I'll leave a link to to that in the show notes to the listeners.
Thank you so much for tuning in to the AI chat podcast.
Make sure to rate us wherever you listen to your podcasts
and have a wonderful rest of your day.
If you are looking for an innovative and creative community of people using chat
GPT, you need to join our chat GPT creators community.
I'll drop a link in the description to this podcast.
We'd love to see you there where we share tips and tricks of what is working in chat GPT.
It's a lot easier than a podcast as you can see screenshots.
You can share and comment on things that are currently working.

So if this sounds interesting to you, check out the link in the comment.
We'd love to have you in the community.
Thanks for joining me on the open AI podcast.
It would mean the world to me if you would rate this podcast
wherever you listen to your podcasts and I'll see you tomorrow.