SESSIONS

LangChain 101 - Virtual Edition: Kickoff 2024 - January 10, 7:01 PM | 8:53 PM

Colin McNamara - 7:01 PM
I can sense it's being recorded, okay. The transcript is started to, starting. Stopped transcription, participants. I see no attendees in this session.

Colin McNamara - 7:01 PM
There's like 46 people that were signed up in Meetup. Copy session link.

Colin McNamara - 7:02 PM
56 people. It's been online. Can't Google. It's my personal account. Sure, it's me.

Colin McNamara - 7:02 PM
Eleanor

Colin McNamara - 7:03 PM
is your user email . . . . . .

Colin McNamara - 7:04 PM
. . . . .

Colin McNamara - 7:05 PM
. . . . . .

Ricky Pirruccio - 7:05 PM
awesome

Colin McNamara - 7:05 PM
. . . . .

Colin McNamara - 7:07 PM
. . . . . .

Colin McNamara - 7:07 PM
. . . . . .

Colin McNamara - 7:08 PM
. . ." We'll ignore that. So we have the chat on the right, so we can chat with each other. You're welcome, Tony. We have some polls that we will be throwing up here and there. I think we can give those to you. We have a Q&A, so please, if there's any questions, cue them up through the Q&A. That'll allow us to go ahead and pick and choose which ones we want to pull back in the session.

Colin McNamara - 7:08 PM
The notes is, I believe, your personal notes, and you can connect them into your Notion and Evernote. If you click takeaways, we have preloaded this with links to our repo, a direct link to the presentation, a link to our meetup, and the direct links to the collab interfaces of the labs that we'll be doing today.

Colin McNamara - 7:09 PM
There is also a transcript that will be a running transcript that will be transcribed, as well as a recording for this. Now, anyone who has registered and attended will have access to the recording, as well as all the transcripts and whatnot. Also, really cool things, there's an API interface to all of this, so in future labs, we'll be able to pull in this cool stuff.

Colin McNamara - 7:09 PM
into our AI microservices for our fun. So, on that, that's the tour of the interface. And if anyone has any questions, feel free to ask in the Q&A. I'm still learning this myself, so, you know, bear with me. Okay, let's stop the share screen, and let's go ahead and share a presentation instead. Okay, I'm going to go ahead.

Colin McNamara - 7:10 PM
And kick it off, make sure that everyone can see things properly. Can I, in the chat, can everyone see this? Give me, like, a thumbs up. Cool. Thanks, Catherine. Okay, so, again, my name is Colin McNamara. I am the organizer of the Austin Lakechain User Group. We are a group of...

Colin McNamara - 7:10 PM
Lakechain enthusiasts based out of Central Texas in Austin, the coolest city in the world, if you ask us. It's more like adult summer camp, but it's full of really great people, really good hearts, really friendly, and a lot of fun things to do, including things like this. The Austin Lakechain User Group, again, is a group of users of Lakechain. It's an AI middleware software that we're going to be going over some basics of how to use today.

Colin McNamara - 7:10 PM
You can find us on our Discord. Discord is to my left-hand side here. People are chatting on that as well. Let's keep our chats in here right now. Feel free to pop in the Discord, say hi. We are on there. There's always a few of us on there active throughout the day. It's really cool to see people posting about what they're doing, asking questions. It's also a place where we come together and plan our both in-person and virtual events.

Colin McNamara - 7:11 PM
Our GitHub is listed here. It's a repository on GitHub. Our software is licensed as Apache 2 and our content is licensed as Creative Commons Attribution. Basically, this means you can use the code for whatever you want, the labs for whatever you want. You just can't sue us for patent infringement if you use that in your own software. As well as the presentations, the content, your free...

Colin McNamara - 7:11 PM
to give this yourself, in your own community, in your own business, in your own consulting practice, whatever. We don't care. We are embracing learning in the open. This is a fast-moving project in a fast-moving field, and there's a power of learning together. We have our Meetup group link, which is there, and the YouTube channel. Thanks, Catherine.

Colin McNamara - 7:12 PM
I forgot to put that in there. We will be producing this latest update and throwing that up on the YouTube. One cool thing about this tool is it should keep a better… Catherine just said the screen is a bit small. How can I make that bigger? I don't know how to make that bigger just yet.

Colin McNamara - 7:12 PM
I will take that feedback though. That's my full screen. I don't know if that works better. Let me see. Hopefully that is a little bigger. If not, we'll go ahead and try to make this better the next time. A little about me. My name is Colin McNamara. I live on the east side of Austin.

Colin McNamara - 7:13 PM
It's kind of the hip, up-and-coming side with the great restaurants and whatnot. A lot of really cool groups of people. For my day job, I am managing partner of engineering and finance at a consumer product development company called Always Cool Brands, dealing with really fun stuff every day. My background is hyperscale cloud engineering and operations. I've spent 25 years of my life.

Colin McNamara - 7:13 PM
Building pieces of, and significant pieces of, as well as operating some platforms that all of you use every day, and I still use every day. My open source story started with Linux back in the late 90s. Moved on to open cloud platforms such as OpenStack, SDN platforms such as OpenDaylight. Progressed into different platforms, and now we're here inside.

Colin McNamara - 7:13 PM
of Lanchain, so open source AI. I'm using Lanchain for business operations, again, so, you know, this is AI middleware. I'll talk a little bit about how I saw value in it, and why I am so eager to continue to learn that. You can find me on the internet at colinmcamero.com. My LinkedIn's under there, too. I post some stuff to Twitter every once in a while, and you can, again, find me on Twitter at colinmcamero.com.

Colin McNamara - 7:14 PM
You can find me on our Discord. I will start with a little story. I will ask you, how safe is your next flight? A couple of days ago, we might have seen the 737 Max issue where one of the door plugs blew off on its way out of Portland, and it kind of highlights aviation safety as something that's really important nowadays. Prior to that, we had

Colin McNamara - 7:15 PM
a lot of talk about fraud in the supply chain around the world. This fraud in the supply chain exposed weakness in the parts market, but it really is representative of fraud in the supply chain all around the world. From our work, we build supply chains, we formulate products, whether it's a juice box, whether it's cookies, whether it's a consumer product like a headphone that goes in your ear, and there are great manufacturers out there, and there are horrible manufacturers out there, and there's everyone in between.

Colin McNamara - 7:15 PM
And so, for my use case, as we're designing private brand and private label products for mainly focus on our customers for retail, grocery, club store, and also influencer agencies, we've been formulating nutraceuticals and different injectables for medical influencers and stuff like that. The work streams that we work in,

Colin McNamara - 7:16 PM
packaging, design, formulation, manufacturing, bulk material sourcing, and so, and we also were, we have to deal with a lot of regulatory environments, so ISO, safe quality foods, we have to create assets for any ingestibles, anything that's FDA-managed, we have to maintain a CAPA process, a corrective and preventive action process, and at times we'll help,

Colin McNamara - 7:16 PM
brands with funding, and so, and that may include selling off a portion of a brand, which now triggers due diligence with the SEC. So for us, the first, first thing, the first time the lang chain peaked its face into my world was a RAG solution for due diligence. RAG is retrieval augment generation. It's basically a way that you can enrich

Colin McNamara - 7:16 PM
an AI agent with documents you present to it. For us, this was really, really powerful. I ended up building it at first because I wanted to make it easier to generate documents for for investors, right? So you need to generate your investment white papers, your thesis, your perspective, stuff like that. What I found was that lang chain provided me a tool to clearly, clearly

Colin McNamara - 7:17 PM
identify where questions were answered thoroughly and where they were not. This identified fraud in my supply chain. Thankfully, we're able to identify the fraud and kick those people out. Kick those problem childs out right as I spill water on my keyboard. Let's see if this blows up our entire presentation or not. Okay, cool. So again, this provided a huge

Colin McNamara - 7:17 PM
amount of value from a free and open source project. And it was actually pretty simple to do. We'll be going over a subset of the code that that I used in our next in person meetup, I believe next week, next Wednesday, as well as a tie. And that was tying in Google Drive into this as well as an example that Ricky will be showing of how to use that using links or templates and like an AI microservice. NetNet link chain

Colin McNamara - 7:18 PM
have protected me from an SEC violation, which is really, really good, because that's bad. But beyond that, it's enabling us to establish provenance. As we have, as we source bulk materials all the way to product ends up on the shelves, link chain as a tool enables us to consolidate information, but also put it put in scope three reporting, and scope three reporting for

Colin McNamara - 7:18 PM
like sustainability issues, carbon, carbon tracking, stuff like that, as well as a bunch of other fun stuff. It's a really, really powerful tool. I am a huge fan. Okay, so what is link chain? Okay, so link chain is an open source, that means it's free. And you can use the source code library for building LLM applications, large language model applications is a fast moving project. It's a fast moving open source project is founded by a really cool

Colin McNamara - 7:19 PM
guy. His name is Harrison Chase. He's at hwchase17 on X. And it is the link chain project is found on linkchain.com and linkchain.dev. We treat this as an upstream project for our user group. And it is it is really amazing. It's moving

really, really fast. And now really cool thing about Harrison Chase, I believe Harvard educated focused on

Colin McNamara - 7:19 PM
robotics, and started writing LM applications and was finding in that when he would, he was rewriting a whole bunch of code as he was writing directly the API is like GPT-4. So the back end of chat GPT, and others. And so what he did is he started he bundled the code that he was using for these common operations by the LM inside libraries, they're consumable for everyone and shared it to the world. What a cool dude. He also

Colin McNamara - 7:20 PM
funded by Benchmark. And I believe his company is valued at $200 million right now. Really cool. The light chain project, it has Python and TypeScript packages, and really is focused on composition modularity. Think of it as a Lego bricks or building blocks that allow us to to build our applications. So we kind of cheat off his notes, cheat off their notes, we're on the team.

Colin McNamara - 7:20 PM
So we'll talk a little bit bit about lane chain key concepts. Let's check my chat, make sure I didn't miss anything that I'm breaking it. Cool, okay. So our lane chain key concepts here. So on the fundamental base of lane chain, and I don't know if you can see my, oh you can, cool. Lane chain itself, and the core project itself.

Colin McNamara - 7:20 PM
Is based on, again, and we're talking about Python versus the JavaScript side, but for the Python side, so it's Python and JavaScript, right? And it has a base, what it does is it composes connectors into models. So models being the large language models on the backend. That you, that effectively compress versions of the Internet.

Colin McNamara - 7:21 PM
That act like smart little humans. It has, the link chain, it has a bunch of different retrievers. So retrieval, ability to retrieve objects out of your Google Drive, out of the Internet, out of a vector store. And there's different ways that you chunk data as it comes in. Has agent tooling. So, and we'll talk a little bit more about this. But the ability for...

Colin McNamara - 7:21 PM
To define the personality of something that thinks and looks like a person, an autonomous agent, that has tools that it can go ahead and do fun stuff with. There's a protocol layer in what's called links chain expression language, which allows you to, in a simple language, similar to a Linux command line, or someone using macOS on the command line, or a terminal, to pipe certain...

Colin McNamara - 7:22 PM
certain chains together, and that's really cool. And there's an application layer, where the chains, the agents, and the executors all tie together. On top of that is a templating layer, where predefined... predefined... link chain... effectively code, you can suck it down and do cool stuff with. You don't have to rewrite a bunch of stuff, you can edit. There's link serve, which is a...

Colin McNamara - 7:22 PM
presentation layer, that allows you to present your AI microservices into your own applications. And then link Smith, which is really a development environment and a debugging engine, that I've been using more and more every day. It's really cool. Let's take a little bit deeper dive. Let's go into some key concepts. Let's talk a little about models. So what's a model? Fundamentally,

Colin McNamara - 7:23 PM
what a model is, is a large language model, is a compressed version of let's say the internet. The newer models are kind of derivative of each other. So you might train a model off a model. But all in all, this starts with what we saw on the internet and in natural language. And with a lot of money and a lot of CPUs, they're highly compressed down, right?

Colin McNamara - 7:23 PM
There's two different types. There's the LLMs, the standard LLMs that you got access to that answer very specifically to queries that you put into it. And there's a chat model. Chat models are optimized to operate in simple natural language. So as you type something simple into it, it may come back with something really, really complex. A really good place to start with. Many of us started with this in chat GPT.

Colin McNamara - 7:23 PM

And many of us, including me, use chat GPT every day. It's a really cool platform. But what we'll be doing with the link chain is using the back-end models behind chat GPT. GPT-4, Turbo, GPT-3.5. And then actually later in this lab, we'll be using a private language model that Karim will be taking us through. So I won't even go out to the Internet.

Colin McNamara - 7:24 PM
So continuing on the core concepts, link chain is an AI middleware that allows you to connect into many, many models. So again, if you're using chat GPT, or maybe you're using the new GPT agents, you're connecting into one model in the back-end, most often GPT-4. Structurally behind chat GPT, it's about...

Colin McNamara - 7:24 PM
eight different models and some routers between it, but I won't get into that. But so link chain supports 20 different LM integrations to all sorts of different models, all sorts of ones publicly available on the internet, as well as ones that you can download privately. As far as 10 text embedding models, text embedding models are the ways when we suck information down on the internet, maybe we suck down a website, or we pull in a PDF document, or we pull...

Colin McNamara - 7:25 PM
in like our internal business processes, and we have a weak wiki, right? There's different ways that you can chunk them up and throw them in the vector store or embed them in a language which the LMs can understand. Right. So it is a highly, highly effective abstraction layer for this, where we saw it's like Sam Altman got a little pissing match with his board a while back, or I guess the flip side, now that we've learned.

Colin McNamara - 7:25 PM
But a lot of us saw the performance of GPT-4 drop, right? And so we had to make other choices for applications. Maybe we switched over to anthropic. Maybe we switched locally to using a llama, right? LangChain allows us to do that very easily, very quickly, and have a composability in our applications. So, continue with our key concepts. LangChain has this notion...

Colin McNamara - 7:26 PM
of prompts. Now, if you've gone deeper into ChatGPT and you started prompt engineering, one of the first things you learn is you need to tell the LLM what it is. Because the LLM has been trained on the entire Internet. And if you're asking for gluten-free vegan cookie recipes, well, you should

Colin McNamara - 7:26 PM
tell it that. And if you're asking for systems engineering commands to ensure that your Docker, if you're trying to figure out your Docker commands and you need systems engineering help, well, there's two different personalities that your LLM needs to have. And so you can help it out by saying, you're a helpful AI that has expertise in DevOps.

Colin McNamara - 7:26 PM
tool chains, or you're a a nutritional scientist that has expertise in formulating healthy recipes that avoid seed oils, right. So in this case, I have an example here of a chat template. And so these are things we can use inside Lang chain where we can preload up these personalities.

Colin McNamara - 7:27 PM
So as we're building our applications, we might want to build an application, in this case, that is a content rewriter. And so if you look in the code here, there's a system message. And oddly enough, this message goes to the system on the back end, right? It's like, hey, you're a helpful assistant that rewrites the user's text to sound more upbeat. Now, if we scroll down, we can see that it's the lm equals chatty chat, open AI.

Colin McNamara - 7:27 PM
So it's the open AI chat interface. And that we're passing all this stuff, the lm chat template, blah, blah, passing this text, which is the human said, I don't like eating tasty things. But on the system side, it says we want to write this text to be more upbeat. So the response, if we look on the bottom, is I absolutely love indulging in delicious treats.

Colin McNamara - 7:28 PM
Now, this is a simple example, but you can use this very, very powerfully. You can define multiple templates, you can reuse them, and these are things that you'll probably use every day. The thing I really want to highlight, while there is some Pythonic code inside of here, to change the function of this application, you just have to change your system message. Right? So really, really fun.

Colin McNamara - 7:28 PM

So, we explored what a Prompt Template was. Now, the next key concept we want to talk about is chains. So, functionally, what we can do is we can chain these Prompt Templates together. So, there's three different types of chains that you're going to run into most of the time. The first one is very simple. It's a simple chain. So, we take an input, we do something to that input. In this case, the input was...

Colin McNamara - 7:29 PM
I don't like eating these things. The output was, I love tasty treats, right? And then you can pass that to chain two. And that input maybe... And that chain maybe make a joke about the text that's coming in, right? And, you know, what did the tasty treat say when it was walking down the road? You know, I don't know. I'm a tasty treat. I don't do these things, right? That can be the output, right? Next, we have sequential chains. So, we can...

Colin McNamara - 7:29 PM
You can pass input to multiple chains. So, you may have multiple simple chains that you use that do a transformation, maybe do log analysis. Maybe you are creating social media blasts out of your blog content, right? Maybe, I don't know, you're doing a bunch of stuff, right? So, in this case, you can have your input pass into multiple chains

Colin McNamara - 7:30 PM
that do transformations and then come back in and you can combine multiple chains together for one output, right? And then the third type, which is really, really cool, is a router chain. So, many of us have found issues with hallucination inside LLMs. There are some large language models that are better than others and some language models that are not as good.

Colin McNamara - 7:30 PM
So, in this case, you provide your input, which may be your query, and the router chain itself looks at the input, and there's multiple ways they can handle this, but the simplest way it actually looks at the words you use and matches that to what is the appropriate chain. So, in this case, we defined a mathematics chain and a quality engineering chain, right?

Colin McNamara - 7:30 PM
So, if we put in saying, okay, I need to make two times two, and then raise that to the fourth power, right? Depending on the LM, it may totally screw up the math. You can define in your router chains, effectively, a personality in that system prompt that says, you are extraordinary, you're a math chain.

Colin McNamara - 7:31 PM
And then you can connect in that chain, you can point it back in that LM prompt, you can point it back to a specific router, or excuse me, a specific model, maybe Wolfram Alpha, or maybe LM Math, which is highly optimized and really good at mathematics. On the flip side, you can maybe throw out a question again, and you're like, oh, hey, I have an instant response.

Colin McNamara - 7:31 PM
That is happening, and my engineers are stuck, right? Here's what's happening, what are my next steps? Boom, that's a quality engineering chain. You know, you throw in reliability engineering for Oracle Cloud for generation one and generation two. Things like this happen all the time during an incident. You can use your router chains to go ahead and pass to not only specific products,

Colin McNamara - 7:32 PM
but if you've augmented those LLMs with internal process documents, runbooks, stuff like that, you can get some really good information that can help your engineers get your customers back online. So those are chains. Let's talk a little bit about retrieval, augment, and generation. LaneChain supports 50-plus document loaders.

Colin McNamara - 7:33 PM
You take in a big document, you kind of have to break it apart, and you have to identify where the concepts are breaking apart. A lot of the fun in RAG is playing with your splitters and making sure you're presenting the right information up to the models, as well as effectively being cognizant of how many tokens you're using, so how much it's going to cost, but also how

Colin McNamara - 7:34 PM
much it's going to cost, and how much it's going to cost to make sure you're presenting the right information up to the models." So I'm going to go through my half-dead plant. I've been traveling too much. It died. And it hops to my Polaroid camera, and then it goes to Alien 1, my astronaut, my rock, my cigars, and to my candle, right? So that's path through space and time. Words and concepts are represented in that

Colin McNamara - 7:34 PM
So I might have the same path. I go from plant, to camera, to rock, to alien, to astronaut, to cigars, to candle. And so the similarity in them are one of the things that we use to identify where similar concepts are. It's a really powerful way of understanding information. So we stick our data in our storage using vectors, right? And then the flip side, we pull it out whenever we need to.

Colin McNamara - 7:35 PM
When we're retrieving it, when we're querying this new type of database, we ask it a question, right? It pulls from the vector store, looks for the relevant, identifies the relevant chunks of information they're hiding in there, passes that to the prompt, the prompt that we gave it a personality, combines that with that LLM, and out comes the answer. That's magic. This is one of the hottest, most fun areas of large language models.

Colin McNamara - 7:35 PM
I'm very encouraged to see all of you doing this. Again, we'll be going over RAG in depth, I don't know. But we're doing our in-person workshop on the 10th, excuse me, 17th, on next Wednesday. So I encourage all of you to join and go deeper. The next concept to talk about is agents. So agents themselves are like little robots. You think of agents for...

Colin McNamara - 7:36 PM
from Mr. Smith from The Matrix, right? They are independent entities, in a sense, capable of thought, more of automatons. They have access to tools. These tools may be the ability to read a database, to write data, to connect to the internet, to pull information, to talk to another agent, to connect and do like a Kafka bus, right? Your tools are reusable and composable, and you can share them with people.

Colin McNamara - 7:36 PM
Pretty cool. Or you can share them with each other. Fundamentally, your agents are a combination of a large language model, the code you write, memory, which we'll go into and we'll explore a little bit more in our first lab, and the tools you provide it to. This is the future. This is what I'm building my business around, right? Ongoing agents that augment mine, my partner's, my employees' work.

Colin McNamara - 7:36 PM
Langserv itself, we talked a little bit of it. Langserv is a really cool trick, thanks to the Langchain team, that bundles Python code and bundles it behind an AI microservice, basically a little web service that you can connect to. What this allows us to do is define agents or functions, AI functions, and have them address...

Colin McNamara - 7:37 PM
publicly or privately, and build a multi-tiered application. The Langserv itself, for your local development, has a command line interface, and makes it really easy to pull down a template for local development, but also to insert into your AI microservice management workflow.

Colin McNamara - 7:37 PM
It's definitely new, we're all running with scissors here, so production is definitely what you're going to use it for. Smaller businesses it's easier, larger businesses you're going to have to have more controls around it, but one really cool thing that Langserv allows you to do is start to get an operational process around managing and deploying, editing, and sharing your Lang chain templates. It allows for super easy templating, super-duper easy templating, both for creating new ones

Colin McNamara - 7:38 PM
but also for downloading ones and editing it. There are so many templates out there and it's growing more every day and it creates this API service layer. We've dealt with microservices a lot over the past decade. It is a really good way of establishing control. It allows us to break our AI code into small components and so we can start to establish a software supply chain for our AI

Colin McNamara - 7:39 PM
frameworks are set up. The ability to have a normal business user pop into a simple web interface and just play and improve and get feedback I think is really really powerful. And so Langsmith is a really fast growing platform for the management and maintenance and improvement of our AI code. I encourage everyone to try it. It provides both functional and non-functional information so you got like late

Colin McNamara - 7:41 PM
There's the ability through Langserv to pull down templates, use a command line interface, or you can just copy them yourself if you want. And then for the ability to look at how your application is performing, to be able to tune it and

improve it. There's a really cool platform called Langsmith that they've been developing. And I'm really thankful for the Langchain team for not only putting together Langsmith.

Colin McNamara - 7:41 PM
And Langserv, but continue to improve the project. You know, I see Harrison checking. I think there's a doc update that's in PR right now that he wrote himself. So, you know, it's a really active project. Really happy. On that note, let's get and do some labs. Okay. First, for those of all, you all that don't have an open AI key, you need one.

Colin McNamara - 7:42 PM
I need to share my browser window, not my thing. So, what I want you to do is to go to openai.com and I'm going to share my window. There it is. Okay, cool. We're going to log in to OpenAI.

Colin McNamara - 7:42 PM
As it's being slow, I'm going to continue with Google for me. Sign in however you want. I'm going to go with my open source account. If anyone needs to contact me, that's my work one. That's my personal. I try to reply in email. We're on slowpoke.

Colin McNamara - 7:43 PM
And oh, they changed the location. API keys. And you can create a new secret key. Right. So, BlinkChain 101. It's a key. Now you all can see this. I'm gonna go ahead and click done and delete that key because people do bad things. Okay. Now they

Colin McNamara - 7:43 PM
have your key. Now I want you to save that. I keep it in a password manager. Feel free to revoke your keys whenever. Note, please don't put your keys in your source code. There's ways of avoiding this. You can use environmental variables. We're gonna be stuffing them inside of a little password manager in Google Cloud. Put them in your source code. If you check them in, especially if you upstream them, they will be

Colin McNamara - 7:44 PM
grabbed by scripts and exhausted and drained. The default for OpenAI is, I think, 10 and 20. So you get a warning at 10 bucks and it cuts off at 20 bucks. So the worst thing that's gonna happen, you're gonna be out of a little money. But, you know, your keys are money. So, you know, take care of your wallet. Oops. Okay, slideshow. Get that going again. Why is this full screen?

Colin McNamara - 7:44 PM
Okay, we're just gonna go forward. Okay. So next, what I want you to do is click on your notebook, which is your intro to LangChain. Now, what I have, if you go into the takeaways, there should be a link that's to lab one, which is a direct link to the cloud. If you're in our

Colin McNamara - 7:45 PM
Show a moment to load up. You can go to the IPYNB file, and this is a Jupyter notebook. For those that are new, the Jupyter notebooks are basically a way to run Python in a notebook. They're used a lot to discuss concepts in data science and machine learning. Really, really powerful. Think of them like a runbook. So there's an open...

Colin McNamara - 7:45 PM
button on here. So if you click the link directly under Lab 1 in the takeaways, it should take you directly to here. For those that are in the repo, just click on that. We are going to get...this is going to open up a tool called Google Cloud. Google Cloud is a hosted interface for these Jupyter notebooks. It is...there's a free tier. I pay the 10 bucks a month, so I get access to some of their GPUs. But the free tier is plenty good.

Colin McNamara - 7:46 PM
What I want you to do here for the first thing is I want you to go File and save a copy in your drive. Now if you don't use Google Drive...if you don't have a Google account, you're one of those, one of the few people. So create one. We're going to go ahead and save a copy in our drive. Now this is something...now you can edit directly in your Google Drive. You can go into our repo and you can clone the repo. You can do whatever if you're super advanced.

Colin McNamara - 7:46 PM
But now we'll have...we'll show that to local copy of my introduction and then we'll get WordTune out of the way and close these release notes. Okay, so now we have a local copy in your drive that you can play with wherever you want. Really, really cool. So in our introduction, we want to do a couple of things. The first thing we want to do is inside this interface...

Colin McNamara - 7:46 PM
See this thing in black here? This is actually a command you can enter. When there's a play button next to it, you can press that and it's going to go ahead and execute that. There's another way you can do it too. You can highlight this so it's a little blinking and you can do Shift-Enter and it's going to execute it. And you can just hit Shift-Enter all the way down. And it'll execute the entire thing. Now also, if you're having any problems with your notebook...

Colin McNamara - 7:47 PM
You can go ahead and do a couple things. So one here... Disconnect and delete runtime is probably the most powerful thing. So this is hosted up at Google. Sometimes things get gummed up. You can throw it away. You can try something else. So let's go ahead and connect. If you see it's connecting to Python 3, Google Compute Engine back in. Boom! I'm connected. How cool is that? Again, if you have any problems, just...

Colin McNamara - 7:47 PM
Here. Disconnect and delete. Okay. Now what we're going to do is we're going to press play here. I'm going to talk you through. So what's PIP? It's a Python package manager. Right? So again, Python notebook. We're saying PIP, please install quietly. So we don't have a lot of log messages. Three packages, four packages here. We've got LangChain, which is the software that comes from the LangChain team. We have OpenAI,

Colin McNamara - 7:49 PM
which we'll actually call this underscore queue, and you can turn it on and off. Now I'm going to go ahead and just delete this because I already have one defined, my OpenAI key. And I'm going to turn this off. Now just for fun, I'm going to turn on my LangSmith stuff. You won't see it right now. We might show it to you in the future.

Colin McNamara - 7:50 PM
You can easily pass a large message to it and have memory. Press play. We see the green checkbox. It means things are good. So next we're going to get the OpenAI underscore API underscore key from CloudSecrets. That is right here. I'm going to turn this off just to show you something. So you don't even have to turn it on manually in that key. We're going to press play.

Colin McNamara - 7:51 PM
Press play here. So from grant access, turn on automatically. How cool is that? So from google.collab, that's what we're in here, right? Import user data. So this is our user data here, right? And the OpenAI underscore API key, the lowercase version that we use in our code, is going to equal the user data.

Colin McNamara - 7:51 PM
of the uppercase API key, which we define here. So we're going to basically bring that in there. If it's there, we're going to say, excuse me, if it's not there, we're going to say it's not found, and if it is there, we're going to say it's found. What do you know? OpenAI key was found in CloudSecret. So we're going to come move a little further down. We're going to create a LangChain object named LLM, and we're going to make it...

Colin McNamara - 7:52 PM
for the OpenAI's chat interface, right? And we're going to pass in, to this chat interface, the key that we just defined. I think I'll keychain here. So press play. It's basically saying there's the API key that we're going to connect to ChatGPT, or GPT-4, back in ChatGPT. Looks like we got some deprecated. It should work. I don't know. Note to self, we should add...

Colin McNamara - 7:52 PM
Specific versions. There's been a refactoring of linkchain code and split between stable and experimental. So hopefully that should work. Okay. Next, we're going to create an object. We're going to create a messages object, right? And what we're going to do is we're going to... It's basically a list, right? Think of it like just writing... If we're going to take our notes in a meeting, right?

Colin McNamara - 7:53 PM
We're going to put the first message in this message object, and it's going to tell the AI that it's an assistant, and we're going to prompt it to be like, hey, you're an assistant, and you need to ask, how may I help you? It's going to tell it to be helpful. Boop. Okay, that worked. Now we are going to go ahead and define a prompt. When we press play here, you notice a window, right?

Colin McNamara - 7:53 PM
Oh no, speaking of healthy things, Stevia, or Zevia, makes a great root beer. I highly recommend it. It's not full of bad stuff, though, I'll kill you. So now we need to do like, okay, what's a prompt? What can we ask our LLM? Hmm. Well,

we can ask it questions about our universe, right? So like, hey.

Colin McNamara - 7:54 PM
Now we're now we're going to append this message, this prompt, content prompt, content prompt. We're gonna say this is from a user. We're gonna pass it a chat message and we're gonna append it to this messages object. Basically, we're just gonna add it to the list. Now we're gonna look at messages and we can see that the first thing we told it was, how may I help you, right? And then we asked how many moons orbit Saturn? The role is user. So now let's take

Colin McNamara - 7:54 PM
okay Saturn has 82 known moons as of 2021. However, new moons are continuously being discovered. So this number may change in the future. Now if you see there's AI messages, there's content, there's some other stuff. We can basically pull just the content from this response. And we can see that Saturn has 82 moons. So this is a very very simple example of how you can

Colin McNamara - 7:55 PM
how we can interact with the application. So now let's take this response and add it back to this little list of notes that we call messages and append it. Now there's this concept when we're making our code of giving AI code memory. Now how we give it memory is we store it in an object, we write it in a file, we throw it in a vector store. In this case the message is object. So let's go back up

Colin McNamara - 7:55 PM
to prompt six. We asked it how many moons orbit Saturn. Say how many of these moons could support life. Go ahead and append that to messages. It will display what we have. Now as you can see, messages has our initial prompt, as we told it was an assistant, our initial question, the answer that came from it,

Colin McNamara - 7:56 PM
an additional question. Now because we're appending this to messages, we're gonna pass this entire thing up to the LLM. This is kind of how you give how you give a large language model memory effectively. Let's get its response and let's pull just the content out for easy. As of now there's no scientific essence that the moons could support life.

Colin McNamara - 7:56 PM
No conditions required for life as we understand it. So stable atmosphere, liquid water, a source of energy, not present moons. However, future missions and discoveries may reveal more about the potential for life." Let's add it again, and I want to do one more follow-up thing because I talked a little about water and I'm really curious. Okay, that's nice.

Colin McNamara - 7:56 PM
Always be nice to your LM, so maybe our professors in the future, how many of these moons are suspected to hold water. Last step again, dink, dink. You see now we have this entire discussion, right? We're giving it context. We're giving it, we're enriching it, right? So this is the kind of game is

Colin McNamara - 7:57 PM
minimizing is how much data can you pass up to your model? How much can it see? Let's pull the response. And several Saturn's moons are suspected to have subsurface oceans or contain significant amounts of water. Enceladus and Titan are the two most notable moons in this regard. There's geysers on Enceladus in the South Pole indicating the presence of a subsurface ocean. Titan, on the other hand, has lakes and rivers.

Colin McNamara - 7:57 PM
Liquid methane and ethane on it. All sorts of good information. So this is a little and we'll go ahead and pin this response. What I wanted to do here was show you a couple concepts. One, look at how many lines of code here. 1, 2, 3, 4,

Colin McNamara - 7:58 PM
6, 7, 8, 9, 10, 11, 12, 13. And how many lines of code are in the container. We'll look at the containers in our later classes. Okay, on that note, I am going to switch back over to this and hand it off to Ricky Perusia. He's going to take us through some really cool stuff of how to create some web services. Ricky, will you?

Ricky Pirruccio - 7:59 PM
Can everyone hear me? Awesome. Let me just share my screen.

Ricky Pirruccio - 7:59 PM
OK

Ricky Pirruccio - 8:00 PM
OK. Hello everyone. I'm Ricardo Borruchio. I go by Ricky. I live here in the amazing Austin, Texas. I come from a background a little different from this group. I have a Bachelor's of Mechanical Engineering. I've worked in manufacturing for all my career. Started out in aerospace,

Ricky Pirruccio - 8:00 PM
transitioned to semiconductor equipment manufacturing with Applied Materials, and currently working in supply chain and actually bringing this knowledge into my work. A little over a year ago I started, I was in the coding boot camp. I was really interested in games and software.

Ricky Pirruccio - 8:00 PM
So I learned JavaScript, React, building full stack applications, as well as interacting with databases, relational and non-relational, Mongo, MySQL, Postgres, and some neat tricks with Docker and GitHub Actions. Really cool experience.

Ricky Pirruccio - 8:01 PM
Hack Reactor galvanized. I was also a tutor at it. And then I actually picked up Python to work in my current job. This was before LinkedChain. But it became really useful for the present moment. So my interest in LinkedChain... I... Manufacturing is cool.

Ricky Pirruccio - 8:01 PM
I think there's a lot more useful things out there, honestly, that we talk about all the time in Discord. So if you're not there, definitely check it out. We have really cool conversations about all the different use cases that we can apply to. On the bottom, there's some links to my social.

Ricky Pirruccio - 8:02 PM
LinkedIn and GitHub. The QR code right here will take you to my LinkedIn. On that note, we will transition to our second lab. And that is... Stop sharing my screen. You should see that on the agenda.

Ricky Pirruccio - 8:02 PM
The lab 2, creating a simple AI microservice with LinkedIn and Streamlit. So, if you click on that collab... I guess I shouldn't share my screen. Sharing back my screen.

Ricky Pirruccio - 8:03 PM
Okay. You should see collab. So, the idea for this lab is to show you how you can build an AI microservice with Streamlit, integrated with LinkedChain. So, Streamlit is a really cool... You can think of it as a front-end framework.

Ricky Pirruccio - 8:03 PM
For working with Python, it makes building UIs with Python, like, extremely easy. It's used a lot by data scientists, machine learning enthusiasts, LinkedChain enthusiasts. And it's really popular nowadays for creating chatbots.

Ricky Pirruccio - 8:04 PM
Here, you can see, we will first install packages. Install pip install streamlet. You just gotta click on this collab play button right here. Don't forget to connect to your runnable.

Ricky Pirruccio - 8:04 PM
So you press this button, I already have it pre-installed. And oh, sorry, this is the wrong one. So we're starting with this is the one for the microservice. Which should it let me see if that link brought you here. Okay, yes, this is correct one.

Ricky Pirruccio - 8:04 PM
You start out with installing packages, and then you just run your application right here. Now pull out that OpenAI API key. Just keep it handy real quick, because we're going to be using it in just a minute. This is our entire application for this chatbot.

Ricky Pirruccio - 8:05 PM
About 40 lines of code. This is all you need to build a chatbot that can interface with an LLM. We're using GPT-4 in this case from OpenAI. And there's really like three parts of this code. You have a sidebar, a message history.

Ricky Pirruccio - 8:05 PM

And then a way to interface with your chat for a user as well as an LLM. So we will dive into the actual UI so you can see exactly what this outputs. Now after you ran this command to write this file, you can run this command right here.

Ricky Pirruccio - 8:06 PM
Uh, that will run, uh, the file on a, uh, on a local host, um, server. Uh, so you, um, after clicking this, you will click this other command right.

Ricky Pirruccio - 8:06 PM
Here, just outputs your IP address, uh, and then there's this NPX command, uh, which we can use to tunnel in our Streamlit, um, runnable into this local tunnel.

Ricky Pirruccio - 8:07 PM
And this is why we need the IP address, so we can put it right here.

Ricky Pirruccio - 8:07 PM
And there you go, this is our UI. Now, the sidebar here is going to ask you for an OpenAI API key. So, you can get your API key, and you can place it here, and then you can start chatting. Let's see,

Ricky Pirruccio - 8:08 PM
cool, so it's working. So, going back to the code, I said earlier, there's three parts to this. You have a sidebar with just two lines of code that we made right here, and then you have the message history, which this is basically the record of our chat.

Ricky Pirruccio - 8:08 PM
And then you have a way to interact with the chat. So, a user has the role of user right here, outputs, inputs a message, and then an LLM right here responds to that message. And we're using the stream handler right here.

Ricky Pirruccio - 8:09 PM
to basically as like middleware between our LLM and our chat. And that's pretty much it. This is all it took to create a chatbot with OpenAI as the LLM layer and Streamlit as our UI interface. It's pretty cool. So,

Ricky Pirruccio - 8:09 PM
now that we have that, we talked about how to build this kind of like an overview out of how to use Streamlit to build this like AI microservice. But let's do a deeper dive on actually like using Streamlit. So, that will be on the second.

Ricky Pirruccio - 8:10 PM
Same thing, make sure you're connected to your runnable and then you can install Streamlit again.

Ricky Pirruccio - 8:10 PM
Then you can press this button right here. It will write this code right here to the Streamlit app.py file. And this is basically to show you how to build a simple, extreme, like, super bare bones, like, Streamlit application.

Ricky Pirruccio - 8:11 PM
So, once you do that, you save this to the Streamlit app.py file. You can go all the way to the bottom here. And we're basically going to execute the same set of commands that we did on the last notebook to output our application.

Ricky Pirruccio - 8:11 PM
And there you go. So, everything on this page is static. There's no user interaction here. We just added a title right here. A header, which is kind of like a title. Some text. More text right here. I see that right.

Ricky Pirruccio - 8:12 PM
We outputted a number. And then we did the same thing down here. In a slightly different way. So, these are kind of like static elements you can think of. Like, I see that text. I see that header. It's kind of how you use Markdown, if you're familiar with it. That's kind of what it reminds me of. I see that right.

Ricky Pirruccio - 8:12 PM
So, you can basically render anything on the page. But, you could also just get the literals straight up and just have them on your code and it will render exactly as you would using sc.write. So, you don't actually need to use sc.write to do this. Now, if we want to make our app...

Ricky Pirruccio - 8:14 PM
it calls widgets. So, it's like sc.textinput or the button, scibar.button. And, they're called widgets because they're interactive elements. At least at South Streamlit, it sort of looks at it.

Ricky Pirruccio - 8:14 PM
Now, here, we're not capturing state. State is basically how you can capture user information or user interaction and persist it between re-renders. So, Streamlit is a server-side render. So, it will be rendered on your server.

Ricky Pirruccio - 8:15 PM
Once that is rendered, it will be outputted to your client. But when your client interacts with it, what happens is that it goes back to the server, re-renders the application, and then outputs it to the client again. And, again, the way we persist user interaction is with SessionState.

Ricky Pirruccio - 8:15 PM
So, pretty important concept for just UI development in general. You might be familiar with that if you're coming from JavaScript world with React. Basically, same thing. So, we will show this on the next section.

Ricky Pirruccio - 8:16 PM
Here you can see our widgets. Again, we can enter name.

Ricky Pirruccio - 8:17 PM
We can enter some text. Press this button, do that. We can increment this counter. You see this object down here. This is basically our session state object. Which is what is getting updated between re-renders. Is our session state. Session state...

Ricky Pirruccio - 8:17 PM
You can initialize it just like a regular dictionary in Python. You can use bracket notation, you can use dot notation. Here you have a widget that you can use to initialize it as well. All you gotta do is just pass it a key. And then you can update your state with these callbacks that you've defined.

Ricky Pirruccio - 8:18 PM
There you go. So, caching. Our goal here is to basically store expensive function outputs. So we don't need to run that function more than once, essentially. It's kind of self-explanatory with the code.

Ricky Pirruccio - 8:19 PM
Here we have some expensive computation, which is like our expensive function. Here we are calling that expensive computation and basically passing in a number. So if I pass one right here, you will

Ricky Pirruccio - 8:19 PM
see this thing run. And that is basically simulating expensive computation. Now, I kind of just cheated here a little bit. For example, I put in a time.sleep function, which basically waits like three seconds before executing the result here. That's kind of to simulate

Ricky Pirruccio - 8:20 PM
And so, if I do two now, it's going to take three seconds to run because it hasn't seen that number yet. But if I go back to one, you will see that it rendered on the screen right away because Streamline just pulled that number straight from the cache instead of running its computation.

Ricky Pirruccio - 8:21 PM
So that kind of just shows you how you can apply this to an LLM response. Maybe you have users asking kind of the same questions over and over again, and you will kind of want to free up resources from your server. You can cache those responses and then output them to your users right away.

Ricky Pirruccio - 8:21 PM
Now this kind of still shows caching. Our next example right here is simulating caching an LLM response.

Ricky Pirruccio - 8:22 PM
The idea is that next example is that this looks like just this is a chat so I wanted to show you how this kind of looks like on within a chatbot application. 35 lines of code is all you need. We have our expensive computation

Ricky Pirruccio - 8:23 PM
we can chat here you will have this little human icon and then you can see that the LLM responds right here with this little robot emoji and I don't know if you noticed but it there was like a spinner on it that took some time let's do another one yo you see a little

Ricky Pirruccio - 8:23 PM
spinner icon right there simulating the expensive computation and let's say hello again ah you can see the little spinner then show up this time because we got our response straight from the cache that's power of caching yes streamlet streamlet automatically creates the emojis but you can actually pass

Ricky Pirruccio - 8:24 PM
and that's pretty much it for this lab I actually recently did something like this for my job I didn't use an LLM to respond to users because the responses were basically like either A or B it's kind of something like that and I was able to use to build a entire...

Ricky Pirruccio - 8:24 PM
kind of just like automating some annoying like data computation stuff that my colleagues were doing and they kind of wow some people too so that was pretty cool but yeah that's basically the power of Streamlit again like not a lot of code super intuitive if you just like

Colin McNamara - 8:25 PM
nothing

Ricky Pirruccio - 8:25 PM
yeah can you build a Streamlit app that isn't publicly visible on the web yeah yeah so whenever you're developing this it will be hosted on your own local machine so you see like a local host forgot what the port number was like 85 oh 8501 right here I'm not sure anymore but yeah 8501

Ricky Pirruccio - 8:26 PM
is the port and that's why we we map that into our local tunnel earlier so we we were mapping the our local collab instance to the worldwide web through that 8501 port

Colin McNamara - 8:26 PM
Yeah, you can run that on your own local infrastructure, on your own laptop or whatever, you can have it privately available. And, you know, as we go through later on in later meetings and are building interfaces for our LangServ applications, it's a great, great use case of how that is completely controlled. Ricky, thank you so much for sharing your knowledge and your positive attitude and contributions to our project. Awesome.

Colin McNamara - 8:26 PM
Thank you so much. I learned so much. Cool. On that note, I want to hand it on over to Mr. Karim Lalani. Karim, you want to talk a little bit about yourself and steal this back and show us some cool stuff?

Karim Lalani - 8:27 PM
Sure, absolutely. Hello, everyone. Thank you for joining us in this joint learning session. Colin, Ricky, and I, we believe in sharing what we learn and learning together. This is something that is ingrained in our philosophy and behind, you know,

Karim Lalani - 8:27 PM
essentially what motivated Colin to start the meet-up and the sessions. I am a software engineer. I've been developing software for over 16 years now. I've worked with a lot of different frameworks and programming languages. I've coded for the back-end, for the front-end, mobile apps, and

Karim Lalani - 8:28 PM
did a little bit of DevOps. And lately, my passion has been sort of getting lost within... Oh, there's a QR code issue. No worries. My LinkedIn should be easy to find. We'll get that fixed, Catherine. Thanks for...

Karim Lalani - 8:28 PM
Thanks for pointing that out. And my interest in, you know, in large language models and LangChain is essentially more on the open source side. I'm mostly interested in, you know, finding and discovering ways by which we can take the power and promise of large language models and artificial intelligence.

Karim Lalani - 8:28 PM
applications built using those models to small and medium-sized businesses. Mostly around, you know, I'm mostly, you know, you'll find me mostly dabbling in the open source side of things, not as much with the hosted services like open AI and Anthropic.

Karim Lalani - 8:29 PM
you know, there's, you know, there's a lot of use cases where you might want to host a model locally. It could be because, you know, you have a niche use case, maybe, you know, the ongoing costs might not be justified.

Karim Lalani - 8:29 PM
For your specific use case, maybe you have data privacy concerns, maybe there's an alignment concern, you know. The answers that are provided by these large language models, they meet a certain bias and maybe those biases, you know, maybe you don't want those biases to creep into your responses to the applications that you build for your personal use or for

Karim Lalani - 8:30 PM
your applications that you build for your organizations or any applications that you build for for everybody else to consume. Maybe you want to hedge your bets against, you know, these changing market conditions, changing situations with, you know, boards of directors deciding one day to get rid of a key figure and

Karim Lalani - 8:30 PM
and then in a couple of days, finding themselves out of a job. So it could, it could be any number of reasons. So that's where, you know, I find, you know, you'll, you'll find me mostly looking into. And the other thing that I'm most, I'm interested in most is to identify ways.

Karim Lalani - 8:30 PM
to deploy these language models. Now, one of the advantages of going with the hosted service like OpenAI is your infrastructure footprint is quite low. You're typically just building, essentially just building small applications that are consuming, you know, they're offloading this computations.

Karim Lalani - 8:31 PM
to OpenAI, to Anthropic, to Google, etc. But when you are hosting your own language model, you have to, there are certain considerations in mind that you have to think about what kind of infrastructure unit, how many GPUs you might need to source. You might have to get creative about different, you know, infrastructure concerns. There might be security concerns.

Karim Lalani - 8:31 PM
You might have to now take on yourself. Having said that, I think let's jump right into the labs. And this lab builds on top of what you've seen so far. The first lab that Colin introduced, where we saw how to build

Karim Lalani - 8:32 PM
a simple application in Python using LangChain that can communicate with OpenAI, and you could communicate with any language model for that matter. But then we saw Ricky put a Streamlit application around that and make it presentable, make it usable. You don't expect your users to log in to

Karim Lalani - 8:32 PM
a Google Colab and run applications the way Colin demonstrated. That was purely for educational purposes. Or, if that is the environment you work in, you're familiar with, then that might make sense. But even then, you saw how easy it is to create a neat-looking application with Streamlit, and it builds on top of the previous.

Karim Lalani - 8:33 PM
work, that previous lab that Colin showed. So, in that same theme, this new lab will show you how to switch out OpenAI with a self-hosted model. Now, the thing about self-hosting a model, models by...

Karim Lalani - 8:33 PM
themselves are inert. They are just binary blobs of files. You can't do anything with them. But, you know, in order to work with them, you need something that's, you know, called an inference engine. OpenAI is providing that to, you know, is providing, letting you use their inference engine through the use of their APIs. But when you're running, when you want to host your own language models locally, then you...

Karim Lalani - 8:33 PM

you need to do it with the help of an inference engine. And for this exercise, I mean, there's multiple inference engines out there. The one that we will use with this exercise is called OLAMA. And we will also be using the Mistral 7 billion parameter model. It's a small model.

Karim Lalani - 8:34 PM
It is small enough to where a free Colab instance will let you run it. So having said that, let's dive right into the lab. Let me go ahead and share our screen. That's the one, actually. Let's do it this way.

Karim Lalani - 8:34 PM
Okay, great, awesome. So again, you'll find this flow very similar to the past presentations, the previous labs. We start by installing Lanchain and Streamlit. In this case, you'll notice an omission here, OpenAI is not included here, because we won't be connecting to OpenAI's APIs. We have the source code here, and I'll go through the source code.

Karim Lalani - 8:35 PM
here in a minute. The next thing we do is, after we save the source code, we download and run the OLAMA binary, and we ask OLAMA to download two models for us. Now, we only need one model for this lab, but to demonstrate that, yes, you can do more than one model, you can host them locally.

Karim Lalani - 8:35 PM
so long as your infrastructure is capable. In this case, we are downloading Mistral and LAMA2, both of them BEAK 7 billion parameter models. We will start Streamlet. At this point, we are already running OLAMA, which is serving us Mistral and LAMA2 through its own API.

Karim Lalani - 8:36 PM
We are then running the application that we just saved here through Streamlet. And then we will open up a local tunnel because Google Colab is not meant to be used this way. You're not meant to run applications using Google Colab. We are just using a hack here. We are running the applications.

Karim Lalani - 8:36 PM
It's backgrounded and then we are doing a local tunnel through this utility called local tunnel, which will give you an easy to access three word URL, where once you go there, you provide the host IP address to verify and then it'll expose the application through it. Now, for the purpose of this video, I'm going to show you how to use Google Colab to run the application that we just saved here through Streamlet. And then we will open up a local tunnel because Google Colab is not meant to be used this way. You're not meant to run applications using Google Colab. We are just using a hack here. We are then running the application that we just saved here through Streamlet. And then we will open up a local tunnel because Google Colab is not meant to be used this way.

Karim Lalani - 8:37 PM
For the purposes of this lab, I went there and I did that beforehand because as it turns out, downloading two models that are four gigabytes each will take a couple of minutes. But let's go through the code quickly before I show you the application again. First thing we're doing, we are bringing in our imports.

Karim Lalani - 8:37 PM
In this case, the one of note is chat-olama. This was what you previously saw as chat-open-ai. So instead of using chat-open-ai, we are using chat-olama. And chat-olama is, again, a chat model interface that is available through Lankchain. We didn't have to install anything other than, you know, you saw that we only installed Lankchain.

Karim Lalani - 8:37 PM
This comes prepackaged with the latest Lankchain library. We are using the human message and AI message schemas for differentiating between the messages that we are passing to the language model versus what it'll return back to us. We'll maintain that in memory.

Karim Lalani - 8:38 PM
We are using a urllib request package and JSON packages, and I'll go over why shortly. And then, of course, we have our Streamlit package here. First call you'll notice here is a getolama model call. It is decorated with a cache resource. Ricky, a few minutes ago, mentioned about caching long-running operations. In this case,

Karim Lalani - 8:38 PM
we will be making calls to the olama API to get a list of models. And we don't want to keep calling that API over and over because that model list is not going to change very often. So we only want to do it once, which is why we've decorated this call with the cache resource decorator. The first time it sees this call, it will run this computation.

Karim Lalani - 8:39 PM
Not a very intensive computation, but again, this is a use case where, you know, if you're running an enterprise level application and you know that the responses are not going to change, but each call is a call over the network, you want to reduce that. So this would be a use case for using a cache resource. We have a stream handler, again, no need to worry about that.

Karim Lalani - 8:39 PM
In the sidebar, we have a text input where we are asking for your local Olama API, defaulted to localhost 11434, and a select box, which is a drop down box with a list of the models. And we got the list of the models from this previous call. Once we have that.

Karim Lalani - 8:40 PM
We have, we also have a button to clear chat history. All that will do is it'll clear out any messages that we've stored so far, and it'll reset it back to the first message with the AI essentially asking, how can I help you? We initialize the state here. And then everything here that you see is pretty much almost, almost verbatim with.

Karim Lalani - 8:40 PM
You know, from the OpenAI chat application that Ricky demonstrated, you know, he showed. The only difference being, we're checking to see if a model is selected. And if it's not selected, then we don't proceed further. We want to make sure that a model is selected before we can chat with them, a large language model. Now, let's see, going with the questions. Why do we need?

Karim Lalani - 8:40 PM
an API call if LLM is downloaded. Okay. And okay. So the reason for that, like I mentioned, the large language model itself is just a binary block by itself. It doesn't, it can't, you know, it's just a file. We are using Olama's inference engine and Olama runs locally as a separate application. In this instance, it's running, we.

Karim Lalani - 8:41 PM
We are running it here in this set of calls. When we're downloading it, we are making it an executable and this Olama serve will essentially run it as a backgrounded application. And one of the things that it does when you do when do Olama serve is it exposes a web service. And the other thing that we're asking you to do is download these two models. The way we will interact with the large.

Karim Lalani - 8:41 PM
Language models is through an API exposed by Olama, which is that's the reason why we are making an API call here. It's still an API call to a local service, but it is still an API call. Does that answer your question, Scott? Perfect. So let's switch over. All right, so this is our application. This is, again, the address of the local service that we're running.

Karim Lalani - 8:42 PM
Now, if you're running in your infrastructure, this could be a separate set of servers. It could be a load-balanced URL with maybe 4, 5, 6 instances of servers with GPUs running Olama on it. In this case, we're connecting to a local instance that is running on the same server as this notebook. This drop-down is showing us the list of models.

Karim Lalani - 8:42 PM
Again, I didn't pre-populate this. This is coming from the API call that was made, and that code is coming from here. It says the select box, and the options for the select are coming from this getOlamaModel function call, which basically takes the URL of the Olama server and makes an API call

Karim Lalani - 8:42 PM
API forward slash tags URL, that is the endpoint that is being hosted. And from there, it gets back JSON response with a list of all the models, and we just extract out the names. And the way to communicate with Olama, because it can serve multiple models at the same time, the way you do that is, you specify that in the chat Olama call. We had a chat OpenAI call, which took some of these other parameters, but in chat Olama, because it can host multiple models, we have to tell it which model we are ... There it is.

Colin McNamara - 8:47 PM
If there's any more questions, feel free to throw them in the Q&A.

Colin McNamara - 8:48 PM
Thanks, Karim, for all the information to share." I learned so much each time, both from you and Rikki. So I want to wrap things up here. So the key takeaways for...

Colin McNamara - 8:48 PM
This is a fast-growing project. As we saw earlier, some of their messages still work. There's refactoring the code base to a core and experimental. It's a fast-growing project. It's full of integrations. There's so many integrations. One of the things I really respect about Harrison and team is that they structured a project where anyone who wanted to integrate with BlankChain could do it quite easily.

Colin McNamara - 8:49 PM
It is incredibly, incredibly, incredibly easy to add stuff in. Karim just added in an integration in, I believe, right before the end of the year. Any of us can do this, and whatever we can do to support everyone, to both writing this code, but also...

Colin McNamara - 8:49 PM
It is incredibly extensible, whether using it for private LLMs, whether using it for foundation models externally, whether using public stuff like GPT-4. It is a Swiss Army knife, and it's growing really fast. It is open. It's open source. It's portable, and it's really accessible. We saw how...

Colin McNamara - 8:50 PM
A few lines of code, it really took to get some basic applications going. So I want to go ahead and extend a warm thank you. And I see some questions here. And so I'm going to answer the first one here. Does LangChain plan to start charging for LangSmith sometime? I do believe so. They're funded by Benchmark.

Colin McNamara - 8:50 PM
I don't know what the tiers are going to be. So that is a question I don't know the answer for. We can go ahead and reach out to Harrison either directly on his Twitter or I'll pop the question in a partner's Slack channel too. And then Blake asked a question.

Colin McNamara - 8:51 PM
Question of AnswerLive, what is the GPU, if any, you are using that returned all that tech so quickly? Both Kareem and I use pay the $10 a month for the basically like a pro version of cloud. It makes it so that it won't kick out your instance will stay OK.

Colin McNamara - 8:51 PM
and longer. It gives you access to GPUs and some stuff. It's kind of. Oh, use the free instance. Oh, cool. Thanks, Kareem. And on that note, OK, cool. I want to thank everyone for your participation. I want to thank Kareem and Ricky for for their work, for.

Karim Lalani - 8:51 PM
I used a free instance for my demo today. Yeah.

Colin McNamara - 8:51 PM
I read their contributions to the project, the contributions to meet up. I want to thank so many of the faces that I recognize from meetups and all these new faces for all the active participation. You know, we're learning in the open. We want to stay cool. You know, we're embracing our our learners minds. You know, we're being vulnerable out here and we're moving forward in this really cool ecosystem. So please connect with us.

Colin McNamara - 8:52 PM
Show up on our Discord, join the conversation, feel free to use the GitHub, our GitHub, teach us internally, teach us externally, show the love, right? Connect to our meetup. We do monthly in person, we do remotes as well. And again, we want to learn and share to grow together. So thank you so much for all your time. Thanks for all the participation. And thanks for collaboration. Also, kind of put up with the growing pains of using these new platforms.

Ricky Pirruccio - 8:52 PM
Happy Thaijan