Python and AI Power-Up Program Offline Class-20250908_113840-Meeting Recording

September 8, 2025, 6:08AM

2h 2m 45s

Ajay Patel started transcription

Tarun Jain 0:04

To have the project section.

So I'll give the boilerplate code on what at all needs to be completed and you guys will just have to fill that particular temp. So if you remember initially when we had a Python session right, I used to give you a template and then you used the code. So similar thing we'll do it here.

Tirth 0:23

Mhm.

Hey.

Tarun Jain 0:26

I'll give you the boilerplate code.

Tirth 0:29

Mhm.

Tarun Jain 0:30

Which is a template.

Or template code.

You will have to fill it.

Tirth 0:40

OK.

Tarun Jain 0:41

So 20% code will be available, 20 to 30% of the code. So 70% of the code needs to be filled and then you need to create a UI. So how to create UI? I'll show that today.

Tirth 0:55
The mic.

Tarun Jain 0:56
You are using stream.
So is this clear? Probably this will be for the first project and I'll give you the boilerplate code and the timelines will be from 12 to 17th, 18th we can have the review.

Tirth 1:13
Mhm.

Tarun Jain 1:14
Cool. So we'll get started with Langraff. So in Langraff, mainly you have to keep four things in mind. One is how to define state, then what are nodes, then what are edges and what are checkpoints.
In most of the cases, checkpoints will not be, but checkpoints are very important. I'll come to what checkpoints are, so I'll define what is state first. So we already covered what is classes, and while defining classes we came across a concept called persistence.
So what is this persistent? Let's suppose I define any attributes.
This attributes is in state management. That means even if you create an object and update this attributes, the value is updated and this value keeps on updating if you have any method which consistently changes the value of the attribute. So what is this concept called? This is mainly called state management.
And when it comes to graph based concept, right graph based concepts.
In graph also what you're supposed to do is you have multiple nodes, so this node is nothing but a root node or you can call any simple tree node.
So what does this node define? This node defines certain logic.
And this logic can be search. Let's suppose I ask any question. You have document from the document you need to retrieve any information. So that is a logic that is a node and the second node is answer generation. So what is answer generation? I get the context.
I give it to the LLM and LLM generates the response. So now basically you have two

nodes, you have node one and you have node two. So this is search.
This is answer.
So now if you look at this node, I will probably add two more nodes and the other two nodes are for evaluation. So now why is evaluation required? Let's suppose I ask a question. I ask a question. I'm getting the response. So there are three different evaluation metrics that we will usually use.
One is answer relevance.
One is context relevance.
And one more is groundedness. So what is answer relevance? You just have to add this formulas which is a slash Q. So is the answer relevant to the question or not? We use this concept as LLM as a judge.
So here how are you evaluating? We are using LLM itself to tell whether the question belongs to answer or not. And next thing is you have Q hyphen Q. So context relevance is in the sense here if you see.
We have context which is been fetched from retriever and we already saw how we are getting the context.
So if you see here I'm testing docs, I'm using DB dot maximum marginal relevancy. I give a user query and then I'm defining K So based on this K, how many text documents do I have? What is the length of this?
What is the length of text docs?
It is 3 right? Which is equivalent to K that you have defined. So now what you're trying to do in context relevance is you are for each chunk that you have chunk one, chunk two, chunk 3.

Ishan Chavda 5:01
OK.

Tirth 5:04
Yeah.

Tarun Jain 5:18
Is this chunk relevant to the question or not? Yes or no, which is binary? And there is one beautiful article that one of the researchers has written on the LLMS are judge.
So what many people try to do is you define values like.
Hey, you define a prompt and you tell LLM that score between 1:00 to 10:00. Now

what is happening is they are using context relevance. They're using answer relevance. Now when you're using LLM as a judge, you're telling LLM.
Based on the answer to question, how relevant is this rank between 1:00 to 10:00? If it is relevant, then probably you will have rank as five or more than five. If it is not relevant, you will have rank equals to 0.
So what these guys says is you should not use ranking based events, rather it should be binary based. So binary based in the sense either yes or no.
So if answer is relevant to the question, it should just generate S and there has to be some kind of justification S plus justification for the answer.
Is this clear? Instead of ranking, you need to have binary based, which is just yes or no. If it is relevant to the question, just mark it as this and there has to be some justification which again is used for tracing and evaluation and then you have groundedness.
So in groundedness what will happen is you actually have a ground to route response and then you are evaluating with the response that the LLM is generating. So this is the actual response. You will have an actual response and then you will also have LLM as response.
And then you are checking whether this is right or wrong. Now why am I telling this? Usually all these three things will happen at same time. So if you look at the answer, now I have three more notes. One is for answer relevance.
One more is for context relevance.
And one more is for.
Groundedness. Now I want you guys to tell me one thing. What are the common variables you see here?
What are the common variables?

Tirth 8:00
We would have the query.

Tarun Jain 8:02
Have a nice query.
Then answer and then. Is this clear? So what we're supposed to do is when we are defining node, we want these three variables to be persistent. So now how will you connect? If you want to connect one node to a different node, this is where you have a concept of edge.

Tirth 8:06
Yeah, yeah.

Tarun Jain 8:24
So edge basically will define how the nodes has to be connected. So nodes is nothing but the logic and edge is nothing but how do you want to connect it and now if you see here.
When you are performing search, you need a vector database and you need a user query. That's it. And once you have the context then only you can proceed to answer. Right now if you look at all these three things, answer relevance is not dependent on context relevance. If you see it, you already have answer.
You already have question.
So these two variables are already present, so question is already there. Answer relevance is coming from this answer node. Now if you look at context relevance, C is already there and question is coming from the user query. Now if you notice here, once you have the answer, you can run all these three parallely.
So whatever arrow marks I'm adding right, those are it on how you need the logic to be like. So in most of the use cases I'll tell some use cases example where it will make much clear. So let's take an example of customer service chatbot.
Customer service chat bot.
And here we just have to focus on two things. One is routing and one more is parallelization. Parallelism all right.
So customer service chatbot. Here what typically happens is let's suppose you have a user query. Now this query can be from different department. So one department is from technical domain and one department is from HR and then one department is from product.
So now what is happening is just consider technical as one collection name, HR as one collection name, product as one collection name. Now all these three things will have their own vector database. So yesterday whatever we did right, I mean in the last week.
You have your own data. You're saving your data inside a collection name. Here also it's the same thing. When you're building a customer service chatbot, you will have different collection name. Every collection name has their own data. Technical person will have their own data. HR will have just business related documents, product will

have.
Anything related to product. Now whenever you have this query, you have to decide for which team does that query belong to? Does it belong to technical team? Does it? Does that belong to HR team or does it belong to product team? So this is where you have routing.
So now how we are routing engine will look like you start with a node, so this node will be our decision making.
Decision making LLM.
And then you will have.
Technical.
I'll just copy this three times.
So this is for technical and this is for HR and this is for product.
And each of them will have search node and answer node. Now this is routing so I will put this as a complete what you call complete arrow mark. Then here what I'll try to do is I want to make this as dotted line.
Just one second, I'll make this dotted line.
So did you guys understand what is happening here? So we are building a customer service chatbot. As soon as you get a query from the user, the first thing is you have a decision making LLM. This decision making LLM will pick which team is responsible to answer this and then you're routing to it.
So when you are routing, if you notice you're not checking HR and you're not checking product, it is dotted line. But when it comes to parallelism, you're going to every single node and you're executing at the same time. This all three things will execute at the same time. When it comes to routing, only one specific node will be picked.
So this is where graphs are very important to understand for different use cases, and in most of the cases you will either encounter routing based application or parallelism. Parallelism will take too much of time. Sorry, parallelism will take time, but the task that you are executing will happen at the same time.
For example, the RFP generation will use parallelism and for evaluation will use parallelism.
Is this clear?
This was just example just to tell how nodes and uh edges works.
Why is this not going?

Tirth 13:36

Let's just skip.

Tarun Jain 13:37

Is this clear? State node and edges? Node is nothing but your logic. Edge is nothing but how do you want to use that particular node? Either it is in parallel mode or it is in routing mode.

Tirth 13:42

Yes.

Tarun Jain 13:55

And in most of the cases the evaluation will happen in parallel and some of the use cases will have routing based application and this is very important. Routing here is where we will use agents.

Where agent is used for decision making.

Pillar, is it clear?

Tirth 14:14

That's clear.

Tarun Jain 14:15

OK, now coming to checkpoints, checkpoints.

Two places itself. One is if you are using memory and 2nd if you are using human inner loop.

So what will happen is let's suppose here after decision making you need to have human to evaluate it.

So here you'll add 1 checkpoint. So after decision making you will have a circle which has a checkpoint. After human reviews, hey you're correct, then only it will go to the specific node. Same here also you can add 1 checkpoint. If the answer is correct, you can add a node if you need a evaluation or not, but typically evaluation will happen on the bunch of.

Hardip Patel 14:40
So.

Tarun Jain 14:59
Data. I guess that is not a good example. Let me give a different example.
Let's suppose there is one use case which is migrating from SQL to Postgres.

Hardip Patel 15:12
OK.

Tarun Jain 15:26
And what you're trying to do is you're using an agent.
And first thing is what you're supposed to do while migrating from SQL to PostgreSQL. The first thing is you need to match the schema.
And once you match the schema, you need to create one architecture diagram.
And this architecture diagram you need a human to review because obviously this is A1 use case where you can't directly depend depend on LLM. You need some kind of human involvement so that you don't make any mistakes. So what will happen now is here you have a node called schema.
After schema you will have a checkpoint with human in a loop will be there which will review if it is right or wrong. Once that checkpoint is marked as yes, then you will have final migration what you call migration function.
Is this clear? Human in a loop is just to verify whether whatever response you got it is right or wrong, and that check is nothing but checkpoints.

Hardip Patel 16:28
OK.

Tarun Jain 16:30
And this is rarely used because most of the people want to just automate their pipeline. But there are still many use cases like migrating from one database to other database where there is some kind of human involved, right? So during those use cases we'll have to use checkpoints.
So the use case that I said right for coding related there I'll add checkpoints.

When we talk about agents, because Langraff is mainly used when you're building agents, RAG, it's very simple. RAG, you don't even need Langraff, but I would just wanted to show the nodes because it's very easy to show node example in RAG.

Hardip Patel 16:58
Yeah.
OK, Sir.

Tarun Jain 17:15
Is this clear? I'll just take one screenshot.

Hardip Patel 17:16
Yeah, go.
Mhm.

Tarun Jain 17:50
OK, so let's proceed. What we can do is we can come to VS code.

Hardip Patel 17:52
So.

Tarun Jain 18:03
I'll create a new folder.
MKDIR.
Rag Langraff.
So what is the first step I need to do?

Hardip Patel 18:16
So.
Activate.

Tirth 18:21
B&B.

Tarun Jain 18:23
OK.


Hardip Patel 18:23
Yeah.

Tarun Jain 18:25
First is I need a virtual environment.


Hardip Patel 18:32
What is sending?

Tarun Jain 18:33
And the comment for virtual environment is.


Tirth 18:35
Python Hyphen M.

Tarun Jain 18:36
This can be anything.
And now I'll just use source.
PPNV bin activate.
Let's use VS code because I want to show how to build UI as well today.
Now what are the code we are supposed to copy?
First thing is we need to download all these packages. I'll just copy this.


Hardip Patel 19:18
Mhm.

Tarun Jain 19:20
And once there are some packages we don't have to install.


Hardip Patel 19:21
Requirements.

Tarun Jain 19:26

Sec dot EXT. So can you tell me which packages we don't need?
We don't need 5 PDFM. Why? Because we already saved our data inside a vector database, so I don't need this loading part.

Hardip Patel 19:37

Um.

Tarun Jain 19:43

Langen community we need. Why? Because I'm using embeddings from here. LLM fast embed we need. Langen quadrant we need OPIC for this particular thing. We'll just skip it, but I hope you understood how to use OPIC.

Hardip Patel 19:48

And.
Yes.
Yes.

Tarun Jain 20:04

OK, so these are the packages. Then what you can do is you can just add stream it. These are the two new lines.
So remaining what you can do is you can just copy it from here only. Just remove this pipe PDFM because we don't need this and also remove OP.
And now you don't need exclamometry mark, so you can remove exclamometry mark.
And you can run all these things at same time. If not best what you can do is you can create requirements.
dot THT file.
There you can just include all this.
You can create a new file, just mark it as requirements dot TXT, then come back to collab. After you are inside a environment variable then only you have to do pip install. Can you see this PPNV? Now how do I get this PPNV after using source? PPNV been activate. You are inside PPNV. Now what you need to do is just type click install hyphen R requirements dot TXT. So hyphen R is for everything that is inside a

particular file and now you can just install.
But do you think this is a right approach to add your requirements dot txt?

21:58
No.

Hardip Patel 21:59
No, no.

Tarun Jain 21:59
No. Why? Because next time when you install, there is no version name for this. So let's suppose your app dot py is there, right? App dot py is working on one specific version, but here requirements dot txt doesn't have any version. This is not the right way to do it. So now what will happen is.

Tirth 22:04
I think.

Tarun Jain 22:17
Once this is installed, what we will do is we will append this particular requirements dot txt.
OK, meanwhile, let's start with the app dot PY. First thing is, let's start with Langraph.
So from.
Langra.
Import there has to be stayed.
Just one second. I want to check from where I'm getting the state graph.

Tirth 22:57
Both.

Hardip Patel 23:11
It still makes sense.

Tarun Jain 23:14
Hi it's from landgraph import stategraph.

Hardip Patel 23:15
OK.

Tarun Jain 23:23
Land graph dot graph. OK, so now if I come back to X Kelly draw.
You have to define start and end. So where is where is my start node?
Where is my start note? So my start note is search.
Then what will happen? Search will go to answer.
And then answer is my end.

Hardip Patel 23:54
8.

Tarun Jain 23:56
So you have to define your start and end node, so I'll come back to VS code.

Hardip Patel 23:57
So.
PDM.

Tarun Jain 24:03
From Landgraf.

Hardip Patel 24:05
And three days I have to solve ATM.

Tarun Jain 24:08
dot graph I need to import.

Hardip Patel 24:11
OK.

Tarun Jain 24:12
Start and end.

Hardip Patel 24:13
Dash VENV Space BPNV jet.

Tarun Jain 24:18
What of it?

Hardip Patel 24:18
No, no, no. I was just fixing.

Tarun Jain 24:23
OK, so everyone are able to install it.

Tirth 24:26
Yes.

Tarun Jain 24:27
OK, so now what I'll do is.

Hardip Patel 24:29
Uh, no. Uh, can you repeat? Sorry, I was in between something.

Tarun Jain 24:30
Please.
OK, so basically what we did was we added some libraries here right in requirements dot txt. After this we ran a command called pip install hyphen R requirements dot txt. This will install everything, but is this the right way to add your requirements dot txt?

Hardip Patel 24:35
Just scan it.
Yeah.
No.

Tarun Jain 24:53
No, we need the version right? So now what we are trying to do is we are just

appending the requirements dot txt and that command is if freeze then greater symbol requirements dot txt.

Hardip Patel 24:56
Oh.
Right.
OK.

Tarun Jain 25:09
Now if you see you have the actual versions.

Hardip Patel 25:13
Hmm.

Tarun Jain 25:18
So pip freeze will tell us in the existing environment variable what packages you have and with the version name. So if I do just pip freeze.
It will show all the packages name with the version. Either you can copy paste. If not you can directly use greater symbol requirements dot txt.

Hardip Patel 25:38
Got it.

Tarun Jain 25:40
OK, so as everyone return these two lines of code.

Hardip Patel 25:45
No, no.

Tarun Jain 25:46
One is I need a state graph.
State graph will define the node and adjust logic.
And start and end will tell me what is the start node.
And then what is the end node?
So from land graph dot graph you can either add all these things in same line.

But I'll keep it separate.
Till here is it done?

Hardip Patel 26:26
Oh, OK.
Yes.

Tarun Jain 26:29
So now what you can do is you can define a state. I will define rack state and inside rack state you have to define specific data type and the data type will be typed dict.
And how will I get typed dict?
From.
Typing.
Import type dict.
And what are the three variables I need now for RAG? One is query and what is the data type of query?

Hardip Patel 27:07
Yeah.
Strength.

Tarun Jain 27:12
Bing.
Then I have context.
So what is the data type of context?

Tirth 27:20
List of things.

Tarun Jain 27:21
It will be list of string. Let me also define list and then I have answer which is string.
So this is how your state will look like.
And then I will define function.
Depth search.
Here you just have to define our state as.

Rag state and what will it return? It will return rag state.
Now here if you notice type dict right? So now what am I supposed to return here?
Here I just have to return.
Context as context.
One second, one second. This should be state.
State of context.
Equals to context. This I need to define. Then you have to return state.

Hardip Patel 28:47
But.

TJ **Tarun Jain** 28:50
So this is similar to what you did in class, what you call class attributes. So how do you define attributes?
This is your object and then this is your.

Hardip Patel 29:03
Then.

TJ **Tarun Jain** 29:03
Attribute which is context.
Same here also what you're supposed to do is state of answer. From where are you getting the answer? Answer equals to LLM dot invoke. We will add a prompt.
And then state dot answer equals to answer.

Hardip Patel 29:23
OK.

TJ **Tarun Jain** 29:23
And then return state.
So you're upending. As of now, there is nothing inside answer string. So what are you trying to do? You're upending it in the answer.
So these are just placeholders. We'll add the actual code and we have already we have already written the code for search and answer.

Hardip Patel 29:44
Yeah.

Tarun Jain 30:01
Till here is done, just the placeholder search and answer.

Hardip Patel 30:01
Mhm.

Tirth 30:05
Yes.

Hardip Patel 30:05
Yeah.

Tarun Jain 30:06
OK, so now what we will do is we will define a workflow.

Hardip Patel 30:08
Sure.

Tirth 30:25
8.
8.

Hardip Patel 30:29
Mm.
Um.

Tarun Jain 30:37
So far what we did was we just defined the state. Now what are we supposed to define next? Node. Node is nothing but the logic which is search and answer. So workflow dot first what do we need to do? We need to add node.
And there are only two nodes that we have. One is search node.

Hardip Patel 30:58
OK.

Tarun Jain 30:59
And 2nd is.

Tirth 31:02
Enter node.

Tarun Jain 31:03
Answer not.
Notice this can be any name. Whatever you have inside your first bracket, that can be any name. Whatever you have second should be your actual functions.
OK.
We will write the code for answer and LLM and prompt, but for timing let's just understand how your Langref code will look like.
This is state. Then we have two nodes and then we just have to create a workflow.
Once you have the edge, then I mean once you have not, the next thing is you have to define edge.

Tirth 31:39
And we will add.

Tarun Jain 31:46
Workflow dot added so I need to start from search.
Then I need to go from search to answer and once I'm at answer then I need to end.
So whatever you use here now should be similar to what first parameter you used if you are using search context.
This has to be search context, search context. This is answer generation, answer generation, answer generation. Is this clear?

Hardip Patel 32:15
So.

Tarun Jain 32:18
Once you do this, you have your graph.

Tirth 32:24
Mm.

Tarun Jain 32:24
That's it. This is how you define.
What do you mean line saying state nodes and it is?
And once you have this workflow then you just have to do print graph of invoke.
And we know how invoke works. We just have to define our query.
And uh, since the data was related to uh the website, I'll just ask mail to contact.

Hardip Patel 33:10
Addage.

Tarun Jain 33:11
So I'll do one thing. I'll keep this inside a variable response equals to.
Graph dot invoke. Now this response has three variables. What is that? One is response of query.
Then print response of context.
I don't want to print context, but this is self understood right? If query and answer is there, even context is there. I will comment this out and then here I'll add answer.
Everyone understood. The simple thing is first you have to figure it out what is our state and the most important part is how you define nodes, right? Because the major thing is in the nodes itself, which has the actual logic. Once you know what is your node, it just is very simple. Edge you just have to define from.

Hardip Patel 34:09
Mm.

Tarun Jain 34:25
Where you have to connect with which particular component. It becomes tricky in adding edge when you're using parallel nodes.

But for that I have a template. I'll show that once we use evaluation.
But this is sequential.
If you define.
The below code.
This is.
Sequential graph.
So you start, then search context, search context, answer generation, then answer generation and then end.

Hardip Patel 35:00
Good.
Mhm.

Tarun Jain 35:05
And then you just have to compile. Once you compile this is now in Lang chain node. I'll say Lang chain object. Notice I'm not telling Lang graph, I'm telling Lang chain object. Why? Because once you use graph you have a function called invoke.
And after you have invoked, you can add your user query and inside response. Now there are three variables. What are those 3 variables or attributes? Query, context and answer. If you want to print query, you can just do response dot query.
If you want to print the response, you can just do response dot answer. You can also print the context, but I'm just commenting out because this has too much of tokens. Which is K equals to three into 2000, which is roughly around 6000.
Let me not tell here if it is done.

Hardip Patel 35:59
Yes.

Tarun Jain 36:00
Is it done?

Tirth 36:01
It is done.

Tarun Jain 36:02

OK, so here now what you can do is we'll come back to this collab.
And.
In this answer generation, if you notice we have embeddings, so let's just copy this embeddings.

Hardip Patel 36:17
Mm.

Tarun Jain 36:20
If last session if you use the VS code you already have it in app dot PY you can just uh copy that.
And then blind.
URL and API key.
Now here what you can do is you can come here and then you can create dot TNV file.
And in this.env file I will define quadrant URL.

Hardip Patel 37:02
I can visit too.

Tarun Jain 37:05
Then quadrant API key.
Then I also need Google API key.
So what you can do is you can create dot ENV file. So when you push the code to GitHub this will be ignored. You just have to add this in gitignore.
So one is quadrant URL equals to your key, then quadrant API key equals to API key, then Google API key equals to your Google API key. Don't add colons, it's just variable equals to value.
Here there is a function called from load.
OK.
Import OS.
Then there is a function called dot PNV load PNV.

Tirth 38:09

That is from.
dot ENV, yeah.

Tarun Jain 38:13
I mean.
It's weird. Why is it not installed?

Hardip Patel 38:20
OK.
Yeah.

Tirth 38:27
Didn't install it.

Tarun Jain 38:31
No, usually some of the packages by default they have it.

Hardip Patel 38:32
dot ENV already in requirements dot text.
It shows in your requirements.

Tarun Jain 38:45
No dot ENV is not there. If you see in BI only have.

Hardip Patel 38:46
No, no Python dash dot no no no Python P.

Ishan Chavda 38:48
Thanks.

Tirth 38:50
Python hyphen dot ENV.

Hardip Patel 38:53
Yeah, it does show that RDM.

Tarun Jain 38:56
OK.

Hardip Patel 38:56
Yeah.

Tirth 38:58
Yeah.

Tarun Jain 39:06
So you just have to add these two lines. It's already there and then OS dot environment. I will add Google API key.
Equals to OS dot.
Get ENV.

Tirth 39:26
Um.

Tarun Jain 39:28
Year of user data.

Tirth 39:29
You don't need to do. You don't need to do the eighth line.

Hardip Patel 39:34
Ah, OK.

Tirth 39:36
Because it will be loading and we'll do that, yeah.

Tarun Jain 39:40
Here we'll need it.

Tirth 39:42
Yeah, here you would need it.

Hardip Patel 39:56
Mm.

Tarun Jain 39:57
And for fast embeds I'll just use from.
Langting.

Tirth 40:04
In ATM biddings, custom bid.

Tarun Jain 40:07
Community dot embeddings dot past embed import past embed embeddings then from quadrant client import quadrant client.
So collection name. What did I use last time?

Hardip Patel 40:25
OK, OK.

Tarun Jain 40:26
It was web pages.
Let me not tell here if it is done. We are just using URL API key and we are defined. If you notice now we are not adding any data, we are just inferencing it.

Tirth 40:33
And then done.

Hardip Patel 40:37
Good.

Tirth 40:43
Hmm.

Tarun Jain 40:45
So we don't have to create collection.
So now you have this DB equals to quadrant vector store. What I will do is I will define this inside state.

Tirth 40:54
M.

Hardip Patel 40:54
Sorry.
OK.

Tarun Jain 40:58
You can define this outside as well, but it's better to add it inside search.
Since I'm using quadrant vector store, I need to import it from line chain.

Tirth 41:11
OK.

Tarun Jain 41:11
Vordent import.
Quadrant vector store. Now what is the next line?

Hardip Patel 41:19
Oh.
No.

Tarun Jain 41:24
I can use directly context equals to DB.

Hardip Patel 41:26
Uh.

Tirth 41:28
TV as retriever.

Tarun Jain 41:31
That is also fine. I'll define retriever equals to DB dot as retriever. Then I have search type.
Search type equals to MMR.
Then search keyword arguments is there.

Tirth 41:53
A equal to.

Tarun Jain 41:58
This is dictionary since it is keyword argument K is 3.
Then retriever has a function called invoke. So now I have context equals to retriever dot invoke.

Hardip Patel 42:10
You.

Tarun Jain 42:19
What do I have inside invoke? I just have to pass state of query.

Hardip Patel 42:23
OK.

Tarun Jain 42:25
So if you want to test this, I'll just copy this here.
I'll come to this retriever. Now if you see this is the same logic DB dot as retriever search type is MMR. Search here arguments is case two. Then here I'll do retriever dot invoke.

Tirth 42:33
Hmm.

Hardip Patel 42:39
So.

Tarun Jain 42:42
Big queries mail to contact.
Yeah.

Hardip Patel 42:50
Open.

Tarun Jain 42:51
So this would generate a document and metadata. Is this clear?

Hardip Patel 42:52
Mm.
Um.
It will take.

Tirth 42:58
Correct Windows 10 dot, right?

Tarun Jain 43:00
So I have to talk to you.
Oh yeah.

Hardip Patel 43:07
Right.

Tirth 43:07
So in when we are doing retriever dot invoke we have to just pass state dot query.

Tarun Jain 43:10
Yeah.
Yes.

Tirth 43:15
OK.

Tarun Jain 43:15
So this is similar to what we did here. Maximum marginal relevancy. You need to pass a user query. So now where is this query coming from? This query is already defined in state, so when you do graph dot invoke you will only give query. Context and answer is generated from node.

Tirth 43:22
Hmm.
Hmm.
OK.

Tarun Jain 43:35
So you just have to do state of query and once you have have context. So now just notice this context is in document and inside document.

Hardip Patel 43:39
And.
OK.
Right.

Tarun Jain 43:48
Page content.
And you have metadata.
But what is our data type?

Tirth 43:56
List of.

Tarun Jain 43:58
List of string. So what do I need to do now? So this is I'll make it as relevant document.

Now what I will do is I will define OK here auto complete which was correct context equals to I will loop.
Through dot in relevant docs. Then what will I do? I will just take.
Yes.
Is this clear what we did here?

Hardip Patel 44:27
We are a little bit behind, but.

Tarun Jain 44:31
OK, I'll just repeat till here you guys are clear, right? What is DB vector store?

Hardip Patel 44:37
Yeah, I'm in like, uh, just let me confirm with OK, OK, yeah.

Tarun Jain 44:38
So what did we do?
So we define embeddings, we define client. So in client we have a collection name called web pages which was already saved. So now when you define vector store, if the data is already saved then it's fine. If it is not saved, we'll just use DB dot.

Tirth 44:53
Oh.

Tarun Jain 45:03
Add documents. So are we supposed to do this now?

Tirth 45:05
2.

Tarun Jain 45:08
No, right. So we don't have to do it. We just have to define client. We have to define collection name where the data is available and for this data you have the embeddings of 768. So whenever user ask any question you have to use the same embeddings to convert it.

Hardip Patel 45:08
Mhm.

Tirth 45:08
No.
Oh.
Got it.

Tarun Jain 45:25
So that embedding is from fast embed.
And once you do this, you have a function called retriever. So what is the first step? First step is R equals to retrieve.

Hardip Patel 45:34
Mm.

Tarun Jain 45:39
So retrieve is nothing but get relevant information for the user query. So you just have to use DB dot as a retriever. The search type is MMR which is maximum marginal relevance and how many K value you need. K is 3 on this line is also same from.

Hardip Patel 45:53
OK.

Tarun Jain 45:56
Last class, I mean last session.

Hardip Patel 45:59
Yeah.

Tarun Jain 46:00
And then when you do invoke and when you pass the user query here my key value is 2 so I will get 2 documents. So when you do the document, this is lunching

document. You have metadata and you have page content, page content, page content.

Tirth 46:15
OK, yeah.

Tarun Jain 46:19
So this will be length of two.

Tirth 46:21
Hmm.

Tarun Jain 46:27
Now what I'm trying to do is if you notice my state, my context needs to be list of string but is it list of string relevant docs. It is document of page content and metadata. So what what am I doing now?

Tirth 46:36
OK.

Tarun Jain 46:42
I know the length of the length of relevant documents is 3, but I need only page content. Page content for sure I know it is string.

Hardip Patel 46:47
Mm.
OK.

Tarun Jain 47:02
So if you look at page content, the data type of page content is string. So I'll just use list comprehension for doc in relevant documents. Just take docs of page content and then save it in a context.
Now have list with three string elements. Is this kind clear?

Tirth 47:23
Clear.

Hardip Patel 47:23
Yeah.

Tarun Jain 47:24
You can either do this or you can do for doc in relevant docs.
And what will this line be like? It will be context dot append.

Tirth 47:36
Thanks.

Tarun Jain 47:38
Doc of page content. Even this is fine if list comprehension is confusing.

Hardip Patel 47:45
Mhm.

Tarun Jain 47:46
Now is this clear?

Tirth 47:47
Yeah.

Hardip Patel 47:48
Yes.

Tarun Jain 47:48
Now the data type of state context is matching with list of STR query is string which you will define when you invoke graph. So here you are using query.

Hardip Patel 47:54

Mm.

Thank you.

Tarun Jain 48:02

A very state of query.

Hardip Patel 48:02

Hmm.

Tirth 48:07

OK.

Tarun Jain 48:07

So let's just confirm if you have completed the search function, let me remove the comments.

Tirth 48:14

Completed.

Hardip Patel 48:16

Mhm.

Tarun Jain 48:18

You can cross it once.

Tirth 48:18

Yeah.

Hardip Patel 48:20

But.

Tarun Jain 48:22

This is the only new code we wrote today. Remaining is same for search.

Tirth 48:27
Yeah.

Hardip Patel 48:27
Mhm.

Tirth 48:35
Langraph is basically we are defining nodes and we are defining the flow of how we want what processing to happen and what should be the output of that process and what is the next step. So I completed this, I completed that and then what next? So it is.

Tarun Jain 48:47
Hello.

Tirth 48:51
You know, like a flow chart.

Tarun Jain 48:54
Correct. So basically many people tell agent to be as workflows.

Tirth 48:59
Mm.

Tarun Jain 49:00
Because this is also just using tools and it's running in a loop.

Hardip Patel 49:01
Yeah.
Mhm.

Tarun Jain 49:06
And that's the reason why many people prefer when it comes to building agentic workflows. Even though they're not pure agentic application, they're just LLM

application. They term it as agents.
So can you build this?

Tirth 49:20
So the application I the application I created for generating policies, it can be rerouted by you know workflow better with Langraph because it is all manual as of now one by one, but Langraph would be much helpful in that scenarios.

Hardip Patel 49:28
OK.

Tarun Jain 49:35
Huh. So initially it was LCL, but now it is Langra.

Tirth 49:43
It's more maintainable as well.

Hardip Patel 49:46
Mm.

Tirth 49:47
Then.

Tarun Jain 49:49
And The thing is, they wrote good documentation only for Langraf. Their langs and documentation. Since it's old, they didn't even care to update it.
But the dog.

Tirth 50:00
Because they're they're running from Langraf as well now, so they won't do it for Langraf.

Tarun Jain 50:05
This is well written.

Hardip Patel 50:06
And.

Tirth 50:07
Hmm.

Tarun Jain 50:08
Compared to lines in document, in lines in document you'll have to search where is and then you have to search the component. You can only search that if you know that particular integration exists or not. If not, it's very difficult to find.

Tirth 50:16
Hmm.

Tarun Jain 50:23
It's like API references you have to check.
But in Landf, it's very systematic. You have subbedding spot and all is required. Then there is also concept guide. Here you'll have everything how to add context engineering, human in a loop where you have checkpoints.
And then where is memory? Memory is there?
So if you see in memory, you have check pointers.

Hardip Patel 50:48
So.
Mm.

Tarun Jain 50:55
Memory is very tricky. Some of their memory doesn't work. We'll cover this till here is it done.

Hardip Patel 51:02
Mm.

Tarun Jain 51:03
Uh, the search function or search node?

Tirth 51:07
Yeah.

Tarun Jain 51:08
OK, so now coming to answer generation, first thing is I'll define LLM.
For a little.
It's Google generative BI.
And I will just copy this LLM equals to Google generative AI. Either you can add this outside function, if not it can be inside function as well.
Model.

Tirth 51:49
You don't need to give parcel or we still need to work with the parcel.

Tarun Jain 51:55
If you need it in Jason then you can do, but here it will return in string only if you see response dot answer so we don't need parsers.

Tirth 51:58
OK.

Tarun Jain 52:05
But if you need it in Jason then we can define.
For rag under done 1% you will not need Jason only for report generation and all you need it.
And then we can define prompt. So for prompt we can use.

Hardip Patel 52:27
System bound.

Tarun Jain 52:28
Chartham.

Hardip Patel 52:30
Mm.

Tarun Jain 52:32
So basically what we usually do is you can create a new file called constants.
Or dot PY. If not, you can do prompts dot PY any any of these two works. I'll do prompts dot PY.
And here you can pass your system prompt.
Make sure it is in capital everything. Even it is small, it's fine, but since we are importing it, we can follow best practices.

Hardip Patel 53:21
Mm.

Tarun Jain 53:21
Should be user prompt.
Just copy the prompts in new file. That's it.

Tirth 53:28
Hmm.

Tarun Jain 53:30
And here what we can do is you can do from.
Prompts import.
System prompt comma user prompt.
So this prompts is a file now.

Tirth 53:48
What is the difference between just from prompts and from dot prompts? Like is it relative or something?

Tarun Jain 53:55
So if you do just dot, it will say from the given directory. In that given directory you already have promised dot PY, so the same thing. So typically how we usually do this is let's suppose you're building a framework and your framework is.

Tirth 54:02
Right. OK.

Hardip Patel 54:10
Alright.

Tarun Jain 54:11
Atyantik is your folder name, so this will be like Atyantik dot prompts. This is just to make sure it is in standardized format considering Atyantik is my root folder which is not, but if you keep it inside Atyantik you can do that.

Tirth 54:18
Oh.
OK.

Tarun Jain 54:32
In most of the frameworks you see it.
Is this clear?

Hardip Patel 54:40
Yes.

Tirth 54:40
Yes.

Tarun Jain 54:41
Now what I need to do is I need to define from langchain.
4.
dot prompts imports.

Chat from template. So we added this three new lines in import statement. You can just confirm if you have added this one is for LLM and two is for prompts and this prompts is nothing but we are adding it inside system prompt and user prompt just to make sure app PY is.

Hardip Patel 55:03
Mm.

Tirth 55:03
M.

Tarun Jain 55:11
Direct only related to code. Anything which is not required which can be used or reused in any other files, you can add it in different file. In most of the cases we'll also have utils dot PY.
So utils dot PY is like if you have any functions which is repeated across multiple files, that time we'll have this file utils dot PY.

Hardip Patel 55:31
Mhm.

Tarun Jain 55:36
M.

Hardip Patel 55:36
Mhm.

Tarun Jain 55:39
Let me come back to the code and this one is done and I will just copy this.

Hardip Patel 55:43
And.

Tarun Jain 55:49
And here now system prompt will be system prompt and this will be user prompt.

Hardip Patel 55:59
Where we need it.

Tarun Jain 56:03
And here you can directly pass from template.
So this one will return content, so I'll just do dot content.
So this is your answer function.

Hardip Patel 56:25
Is your order.
Mm.

Tarun Jain 56:31
And if possible for the projects right, if you can use LLMS from Azure, it will be much useful because Gemini will have limits and all.

Hardip Patel 56:38
Um.

Tirth 56:40
Mm.

Tarun Jain 56:41
And does your?

Hardip Patel 56:41
Anything that.

Tarun Jain 56:44
Where you can use models from open AI and GPT 4O is very good GPT 4O and GPT 4O .1.

Tirth 56:53
Yeah, with Azure that is good. I tried with GPT 5, it never worked better. I think I I also

wanted to ask we faced an issue with Azure. It called some issue with research parameter not found or something like that. So it's not supporting the research or something.

Hardip Patel 57:09
Reasoning, reasoning parameter, reasoning parameter.

Tirth 57:11
Which one?
The reasoning parameter. Yeah, reasoning parameter.

Tarun Jain 57:17
Which model are you using?

Tirth 57:19
Uh model uh open open A uh so chat GPT5 uh GPT5 chat.

Hardip Patel 57:20
Open AI.

Tarun Jain 57:27
I mean, are you using Azure Open AI client?

Tirth 57:29
Uh, Azure.

Hardip Patel 57:31
They're opening again.

Tirth 57:31
Yes, yes.

Tarun Jain 57:33
OK, so once I'll send you that parameter, what you need to use for reasoning tokens, you can add that.

Hardip Patel 57:35
OK.

Tirth 57:38
OK.
OK, OK.

Tarun Jain 57:41
So this is your line right from Open AI import Azure Open AI client and then you are defining client equals to Azure Open AI. You're using like this or are you using from Land chain?

Tirth 57:47
But.
But.
No, no from this one only from open AI.

Tarun Jain 57:56
OK, OK, I'll send it. There is one.

Tirth 57:58
OK. Thanks.

Hardip Patel 57:59
I'm using Azure chat open AI. I think for this I talks.

Tarun Jain 58:03
Anything works. So this is similar right? Here if you see you have two things. One is you have chat, Google generative AI, you also have Google generative AI. One function varies but defining client is same.

Tirth 58:14
Mhm.

Hardip Patel 58:15
And.

Tirth 58:18
Mm.

Hardip Patel 58:18
Um.

Tarun Jain 58:19
So now after this how you define is different.

Tirth 58:23
OK.

Hardip Patel 58:24
Yeah.

Tarun Jain 58:25
So if you're using Azure Open AI probably will not have chat completion, you will just have completion.

Hardip Patel 58:30
Yeah.
Right.

Tarun Jain 58:37
OK, so is this 3 lines and answer generation done?

Hardip Patel 58:41
Just the song.

Tarun Jain 58:43

OK, now you see workflow is done, then graph is done. So if you print graph you will see an actual graph. So what I will do is I will copy this entire code.

Hardip Patel 58:55
It's not.

Tarun Jain 58:55
And I will show it in uh.

Tirth 58:58
Linker.

Tarun Jain 58:59
No, I'll show it in VS code once. Sorry collab what this graph will return.

Hardip Patel 59:07
Show me answer.
When is Quidlal in from template?
OK.

Tirth 59:17
You'll also need the user prompt and the system prompt.

Tarun Jain 59:36
It.

Hardip Patel 59:39
I think enlist.

Tirth 59:47
Hmm.

Tarun Jain 59:48
So now if I print graph you will see you have start, you have, you have answer generation, you haven't. So this will help you to debug how your graph looks like.

Are you using sequential or are you using any parallel node? This will help you to debug how your workflow will look like.

Tirth 59:51
Wow.

Hardip Patel 1:00:03
Yeah.

Tarun Jain 1:00:07
And when you're in the meetings, if you have code, you can just show this workflow to the customers as well. Like this is how we are usually executing it. So now if you look at these keywords, it is what you have defined.

Tirth 1:00:16
Hmm.

Tarun Jain 1:00:22
And search context answer generation. This has nothing to do with the keywords that you're using it for function. So this function is different. So you can use different variables, but usually what we do is we keep it same.

Hardip Patel 1:00:24
OK.
T b,

Tarun Jain 1:00:38
This was just to show that you can have it any name. This is similar to environment variables.

Hardip Patel 1:00:47
OK.

Tarun Jain 1:00:47
You can name this anything. You can also have this as just URL small letter URL. But

still why am I doing it? I'm keeping quadrant URL. This has to match with my actual environment variable.
Where is app dot PY? So if you see here when you do Google generative UI Google API key it has to match. That's the reason why I'm using Google API key exact same. So here also it's the same thing so instead of search context.
If you want to keep it meaningful, you can keep it meaningful.
So instead what I will do is here I can change if you want to keep it meaningful and then I can match that here.

Hardip Patel 1:01:31
Those.

TJ **Tarun Jain** 1:01:33
Is this clear?

Hardip Patel 1:01:34
Hmm.

TJ **Tarun Jain** 1:01:36
Let me back so that I don't get any error.

Hardip Patel 1:01:37
Yes.

TJ **Tarun Jain** 1:01:41
OK, now what you can do is you can directly run it.
It's an app dot PY.
First it will load the embedding model because I'm not loaded embedding model before. So that is where you can add that in utils. So utils what you can do is we can define a function called hold start.

Hardip Patel 1:01:59
Oh.
Mhm.
OK.

Tarun Jain 1:02:12
And inside this you can define your embedding model which is a fast embed. Because this is just one time process, right? You don't have to load this model files again. So what you can do is you can push on utils, then you can run app dot P1.

Hardip Patel 1:02:27
Mhm.
In line 83, no problem, no query, no query.

Tarun Jain 1:02:29
So.
Then you can define pollstar.

Hardip Patel 1:02:38
Delete.

Tarun Jain 1:02:39
OK, this will take I guess my Internet is low. So what I'll do is whatever the below code till graph I've copied. If you list till graph it is copied. Now what I'll do is I'll just.

Hardip Patel 1:02:43
Give context.
OK.

Tarun Jain 1:02:55
Add this last file lines.
Response equals to graph dot invoke. You can add your query. This is your dictionary and make sure your query is your class attribute which I've already defined in the state query.
And now this will have three objects.

Hardip Patel 1:03:22
Yeah, I'm having this.

Tirth 1:03:24
Document object is not scriptable.
She made it.

Hardip Patel 1:03:28
Yeah, yeah.

Tarun Jain 1:03:28
Context is dictionary context dot open doc page content.
Do you make any mistakes?

Tirth 1:03:41
I think when we're getting documents from.
Retriever.

Tarun Jain 1:03:46
Uh, do one thing, just change this to dot page content.

Hardip Patel 1:03:55
We can OK.

Tarun Jain 1:03:56
Here instead of a square, do dot and here.

Tirth 1:03:57
OK.

Hardip Patel 1:03:58
Uh.

Tirth 1:03:59
Mm.

Tarun Jain 1:04:01

What you can do is one second. We define system prompt, we define user prompt.
Where are the input variables?

Tirth 1:04:10
It would be passed automatically, no?
Like uh LLM invoke and OK then we have to pass the variables for query and context.

Tarun Jain 1:04:20
Correct. We didn't pass that.

Tirth 1:04:22
Yeah, yeah, we have to pass the query in context there.

Tarun Jain 1:04:23
Text Oji.

Hardip Patel 1:04:25
Yeah, context we have to pass no.

Tirth 1:04:27
Query in context both. So with invoke we will go with input config.
From template.
OK.

Tarun Jain 1:04:45
OK, then you're using LCL, so LCL was doing it. Here, let's do one thing.

Hardip Patel 1:04:45
No.
Yeah, we were using LCL back then.

Tarun Jain 1:04:53
Here what you can do is you can use F string for user prompt. So I'll do format.

Hardip Patel 1:05:04
Um.

Tarun Jain 1:05:04
So here we have to use an actual variable because if you see we have two input variables here.

Hardip Patel 1:05:11
Mm.

Tarun Jain 1:05:13
Insert prompts. One is query and one more is.

Hardip Patel 1:05:17
Context.

Tarun Jain 1:05:19
Context I'll do format.
I don't know if format works or not.

Hardip Patel 1:05:24
Yeah, it does work right.

Tarun Jain 1:05:29
So I'll define context here. First define context equals to.
Space dot join.
State context. So this is the first line I've written. I made only two changes. Can you notice the two change? The first change is I used doc dot page content. This is the first change and the second change is I did context equals to.

Hardip Patel 1:05:52
OK.

Tarun Jain 1:05:57
Space join state context. So now what is happening here? I'm converting list to string.

Hardip Patel 1:06:03
No.

Tarun Jain 1:06:07
Is this clear?

Hardip Patel 1:06:08
No.
Yeah.

Tarun Jain 1:06:09
And when I do format.
I'll do context equals to context.
And then query equals to state dot query.
Now why is this format used? This is similar to F string. So if you notice here I have context and query. I just have to replace this context with an actual information and query also with actual information.
So I'll just copy these two lines, two functions.
In VS code.
I made only two changes. One is this thing is changed. Instead of square bracket I made it dot.

Tirth 1:06:50
Must be prompt when you was your list.

Hardip Patel 1:06:54
Same for me.

Tirth 1:06:55
Input type class length chain prom dot chat chat template must be a prompt value. List of base messages.

Tarun Jain 1:07:11
You got error again.

Tirth 1:07:14
Yeah.

Tarun Jain 1:07:15
One second.

Tirth 1:07:15
Must be a prompt value that is the error.

Hardip Patel 1:07:15
Yeah.

Tarun Jain 1:07:22
Is this better?

Tirth 1:07:24
Yes, that is correct.

Hardip Patel 1:07:26
But.

Tarun Jain 1:07:28
OK.

Tirth 1:07:28
Invalid input type.

Hardip Patel 1:07:29
M.

Tarun Jain 1:07:32

So I'll just copy this.
Paste it outside.
Context. I'll just give some dummy data. This is context.
This is very.
Now if I pass this to LLM.
dot invoke.

Hardip Patel 1:08:30
Mm.

Tarun Jain 1:08:41
How did we define earlier?
If not, let's do one thing. Let's directly remove the prompt template only.
I'll directly use prompt equals to system prompt plus user prompt.
dot format.

Tirth 1:09:07
Template. I think we're good at from template. We have to do prompt template dot formatted messages and then we have to give the formatted prompt.

Tarun Jain 1:09:20
From template.

Hardip Patel 1:09:26
Oh.

Tarun Jain 1:09:46
Right.

Tirth 1:09:49
This is.

Tarun Jain 1:09:52
OK, let me try this here.

Hardip Patel 1:09:55
Um, OK.

Tarun Jain 1:10:04
Now here I just have to run.
Yeah, this work.

Tirth 1:10:10
Mhm.

Tarun Jain 1:10:12
Response equals to graph dot invoke.
Ready.
This mail to on.
I hear any response of answer.
I don't have answer. Oh, we have to return state.
8.

Hardip Patel 1:10:49
OK.
Yeah.

Tirth 1:10:54
They have the answer with me.

Tarun Jain 1:10:54
I took query, I should get the response.
Query is there. Context is there.

Tirth 1:11:01
Hmm.

Tarun Jain 1:11:02
Answer is not that because answer we are returning answer it should be state.

Uh, now this works.
So guys two changes. The first change is the same thing that you have done. You have to use doc dot page content. This will solve the issue of document is not.

Hardip Patel 1:11:11
Yes.

Tarun Jain 1:11:22
Subscriptable. What was that error? Subscriptable. So have you guys changed this? Can you just check it?

Hardip Patel 1:11:29
So.

Tirth 1:11:31
Yes.

Hardip Patel 1:11:31
Yeah.

Tarun Jain 1:11:32
In search function now in answer function define context. Now why are we defining this context? It is in list. I'm converting that list to STR and then formatting it inside prompt.
Let me get back from template.

Hardip Patel 1:11:49
Thank you.
Hmm.

Tarun Jain 1:11:55
And once you define your prompt template, when you do invoke you just have to do dot then define a function called format messages.
And instead of returning answer, just return state. This should work.

Tirth 1:12:04
M.

Hardip Patel 1:12:08
OK.

Tarun Jain 1:12:14
You can just cross check the answer function if it is correct.

Hardip Patel 1:12:16
But.

Tarun Jain 1:12:19
One is this format message and one more is we have to return state instead of answer.
Why? Because if you look at the return data type, the type dict, it should be ratched.
Now if I print this.

Hardip Patel 1:12:46
Context.

Tirth 1:12:49
It's working fine. It's working good.

Tarun Jain 1:12:53
The only issue with it bit slow because the embedding it will define it again.

Hardip Patel 1:12:54
Thank you.
M.
Mm.

Tarun Jain 1:13:08
Is this clear? And make sure you always have this diagram in place.

Tirth 1:13:13
Yes.

Hardip Patel 1:13:14
OK.

Tarun Jain 1:13:14
Now if you want to display it, there is a package called from ipython.

Hardip Patel 1:13:15
No.
Yeah.

Tarun Jain 1:13:23
Import display.
And you can just do display dot.
Display. There is something related to mermaid. This is a mermaid flow.

Tirth 1:13:37
Yeah.

Hardip Patel 1:13:38
And.

Tarun Jain 1:13:40
Oh.
Wait, does it give auto complete here from I pipe?

Hardip Patel 1:13:44
Good.
Not coming out.

Tarun Jain 1:13:49
There is one here graph dot mermaid something.

Yeah.
Get graph. So if I print this I will get this graph.
Now from ipython dot display.
Import display.
Hold down. Wait, why am I wasting time?
Huh, these two lines.
From ipython dot display you will have an image display then same function that I used here. You will get get graph. Now this has to be an image. You use a function called draw mermit PNC so you can just copy this too.
And get it in VS4 as well.

Tirth 1:15:10
And I Python is inbuilt.

Tarun Jain 1:15:13
Oh, we'll have to install it. This is interactive.

Hardip Patel 1:15:14
No.
Ingredient Jupiter Diamond.

Tarun Jain 1:15:26
OK, let's come with the UI part. I'll just quickly show how Streamlit works. So can you create a new file and you can just keep it as utils.py or inference.py?

Tirth 1:15:40
Yep.

Tarun Jain 1:15:42
I'll just comment this code.
So from I can do import streamlet as St. Now this is very simple if you want to use streamlet. If you know English you can build a web app. So what I'll do is I'll just define title.

Hardip Patel 1:16:00
So.

Tarun Jain 1:16:02
And I'll keep it as at the antic chat bot and if I need any sub beddings I can do sub bedder.
Web pages or I can keep it chat with web page.
And if you need any simple message, you can just have.
Streamlit dot right? This is a simple message.

Hardip Patel 1:16:29
OK.

Tarun Jain 1:16:35
If you need any file upload option, you can have file equals to St. dot.
File upload and if you open the bracket you will see what and all you need to add and you will also have examples. So if you see upload images, if you want to allow multiple you can also allow that. There are three examples if you see.
And in most of the streamlet, this thing right, the documentation is very well written, even their code. If you see all this thing is a doc string, how detailed doc strings have written.

Tirth 1:17:08
Um.
Mhm.

Hardip Patel 1:17:11
Um.

Tarun Jain 1:17:13
So here if you see this is one example, choose a file and then you have second example if you want to allow multiple files which is true and then it should be only type CSV so it won't accept anything apart from CSV. So I'll just copy this one.
And here I'll keep it PDF.

Tirth 1:17:43
Yeah.

Tarun Jain 1:17:43
And then streamlit dot sidebar dot. If you want anything in markdown, you can add the markdown syntax.
I'll just add about us.
And I'll make it triple string because I want multiple lines.
About those.
Let me not tell here if it is done. One is title sub bidder right? Upload files. We don't need upload files, but I was showing these are widgets.

Hardip Patel 1:18:20
OK.

Tirth 1:18:27
Hmm.

Tarun Jain 1:18:27
I can also have button.

Hardip Patel 1:18:28
Mhm.

Tarun Jain 1:18:30
Button equals to St. dot button.
Submit. If you click on this, it is true. If you don't click on this, it is false. So which is basically a boolean.
So what is the data type of button? It is boolean.
If you click on that, it's true. If you don't click, by default it is false.
You can try to print button.
And now what you need to do is come to terminal, just do stream lit run utils dot PY.
So this has to be a file where you have your actual stream lit code.
I hope everyone got app dot PY response, right?

Hardip Patel 1:19:23
Yeah.

Tarun Jain 1:19:24
OK, so now we'll just try to build a UI for it.

Tirth 1:19:27
But we won't have stream lit in the CLA, no? Or will it be there by default?

Tarun Jain 1:19:31
No, it will create a UI. It will give you a low.

Hardip Patel 1:19:33
We install Streamlitin pip.

Tarun Jain 1:19:36
So this will give you a local location.

Tirth 1:19:37
Yeah.
OK.

Tarun Jain 1:19:40
You just copy the local URL.
So this is how it will look like.

Hardip Patel 1:19:50
Mm.

Tarun Jain 1:19:50
You don't need HTMLCSS anything.

Hardip Patel 1:19:53
Yeah.

Tarun Jain 1:19:53
This is a web app for data.


Tirth 1:19:54
Hmm.


Hardip Patel 1:19:58
We don't need PS. We don't need Java.

Tarun Jain 1:19:59
So.


Tirth 1:20:00
We don't need anything, Evy.


Hardip Patel 1:20:04
Yeah, Python is more than.

Tarun Jain 1:20:05
So can you see false false? So now if you click on button it is true.


Tirth 1:20:08
Yes.


Hardip Patel 1:20:09
Yes.
Yeah.


Tirth 1:20:13
But then what about multi pages? How? How do you do multi page from this?

Tarun Jain 1:20:18
That you can create. So here what you can do is you can have something called as pages and inside pages you can have what multiple page you need.

Tirth 1:20:25
That's sure.
OK. And then we have to give stream lead run pages.

Tarun Jain 1:20:32
No, you can just do main app and if you want to do that you can just add slash that particular page.

Tirth 1:20:38
It's OK.

Tarun Jain 1:20:41
So this is something that we usually do for demos.

Tirth 1:20:44
I see.

Tarun Jain 1:20:45
For POC or anything.

Hardip Patel 1:20:47
Mhm.

Tarun Jain 1:20:51
No.
2.

Tirth 1:20:52
But lot of effort.

Hardip Patel 1:20:54
Yeah.

Tirth 1:20:56
And there is a deploy button at top. What does the deploy button do?

Tarun Jain 1:20:59
Uh, you can deploy this app as well.

Tirth 1:21:04
That's it.

Tarun Jain 1:21:04
So you just have to open.

Tirth 1:21:07
But it will also take the Python code along with it.

Tarun Jain 1:21:10
Yeah, so if you see there are all these files that I've deployed, data analysis agent, GSoC agent. And if you look at their code, it is just two files, just two files code and those two files are app dot PY and requirement dot TXT.

Tirth 1:21:11
OK.
OK.
That's it. That's it.

Tarun Jain 1:21:26
If you have any supporting files you can also upload it, but in most of my code I just have app dot PY and requirements dot TXT because we quickly build this and deploy it.

Tirth 1:21:27
OK.

Tarun Jain 1:21:39
We'll also try to deploy this chatbot that we built.

Tirth 1:21:44
But after adding a file you have to add events and everything like on submit button what would happen? So it would all be there like it would be a syntax right?

Tarun Jain 1:21:52
Form there you can use form if you want. Inside that form you can have upload files and button. So it's just simple English. Whatever you need right? You can just do streamlit dot and you will have the options.

Tirth 1:21:56
Sure.
I see.
Hmm.

Tarun Jain 1:22:06
So if you need what you call radio button, you can have radio and just open square bracket. You'll find examples. This is the best thing about streamlet. They have too much of examples in the docs.

Tirth 1:22:12
Mm.
Hmm.
I think.

Tarun Jain 1:22:21
Which not many documentation are.

Tirth 1:22:23
I think with this kind of documentation, even the code Gen. will generate the code for you if you write it, you know, basically, yeah, OK.

Tarun Jain 1:22:30
And it will be crazy because there is no other context.
So now if you see I'm defining this genre.
And if you need emoji, you can use this emoji colon and then rainbow.

Tirth 1:22:44
M.

Tarun Jain 1:22:49
So now if I come back.
If I refresh, you have comedy, drama, documentary. You can click on documentary and then you can click on submit.

Tirth 1:22:58
Oh.

Tarun Jain 1:23:15
And there is one more. So which one is that drop down SD dot?

Hardip Patel 1:23:21
Where do you go?

Tarun Jain 1:23:25
Select your favorite color.

Tirth 1:23:26
M.

Hardip Patel 1:23:31
That's it is.

Tarun Jain 1:23:32
Select box, select box.

Tirth 1:23:33
Select. Select. Yeah.

Hardip Patel 1:23:34
But.

Tirth 1:23:39
Um.

Tarun Jain 1:23:40
So for Streamlit, the only programming language we need is English.

Tirth 1:23:44
Um.

Tarun Jain 1:23:45
Like if you need button, just stream the dot button.
We just have to start with St. dot.

Tirth 1:23:55
Yeah.

Hardip Patel 1:23:55
I'm gonna do it.

Tarun Jain 1:23:57
And there is this picker as well. So let's suppose I want to define top K value and I want to give flexibility for user. I'll just use slider.

Hardip Patel 1:24:03
Yes.
Why did you do not?

Tirth 1:24:07
Mm.

Tarun Jain 1:24:08
Define and then range.

Tirth 1:24:10
Yeah.

Tarun Jain 1:24:13
So the range is 02.

Tirth 1:24:15
125 with 25 steps, yeah.

Hardip Patel 1:24:19
Is there any like system?

Tarun Jain 1:24:19
So I'll start with 0, go till 10.

Tirth 1:24:20
This is like range. Yeah, this is like the range.

Tarun Jain 1:24:27
So if you see, it will be there.

Tirth 1:24:31
Nice. Where did go to seven? Oh, how would there be 7? I saw seven in somewhere.

Hardip Patel 1:24:33
Um.

Tarun Jain 1:24:38

OK, so basically this two is the starting point.
So if I make this 4/4 is the starting point, it's not step, it's starting.

Tirth 1:24:45
Atcha.
2.

Hardip Patel 1:24:48
The default value.

Tirth 1:24:48
Yeah, OK. OK.

Tarun Jain 1:24:51
So this I can keep it too. This I'll keep it 7 because more than seven you shouldn't use it for rag.

Tirth 1:24:57
Mm.

Tarun Jain 1:24:58
Then four is the default one.

Tirth 1:25:00
Before, before 12.

Tarun Jain 1:25:02
Now this is ideal.

Hardip Patel 1:25:03
So.

Tirth 1:25:04
M.
OK.

Tarun Jain 1:25:08
Is this clear? Clear. OK, so now what we'll do is whatever code you have written here, we will just modify here. Just remove the print statements.

Tirth 1:25:10
Yes, yes.
No.

Tarun Jain 1:25:21
And also remove the response because I don't need the response. This response has to go inside a stream that attribute.

Tirth 1:25:29
M.
Hmm.

Tarun Jain 1:25:33
Just come back to the same code app dot PY.
And here let's just define online which is from stream lit.
Sorry, not from import stream dot St. and I will repeat these two lines which is St. dot title.

Tirth 1:25:53
Yes.

Tarun Jain 1:25:58
If you want, you can create a main function and then you can define all this.
So inside my main function I will add the graph node.
And now I'll just define query equals to St. dot text input.
Enter your query, get response. So what I'm trying to do, I need a entry box.
I need a entry box. With the entry box I just have to tell the variable called enter your query. Once you enter the query I have a button which says get response. We can add. We can make this more stylish by adding something called as SD dot spinner.

Tirth 1:27:01
Hmm.

Tarun Jain 1:27:01
You can have generating response.
This we can do after you click on button.

Tirth 1:27:09
OK.

Tarun Jain 1:27:11
And then state equals to graph dot invoke query whatever user has entered query and then we just use St. dot right? But I'll use markdown.
State dot answer and then just call main function.
We can also have custom HTML and CSS if you want, because most of the people also want to have their own style design. So during that time you can also define your stream. I mean HTML and CSS within streamlit.

Tirth 1:27:57
So in this line graph we can also have.
A state of previous chat or chat history as a list of strings or a dictionary list of the you know list of dictionary with question answer like what we communicated and then we can also pass it as a context in the same state.

Tarun Jain 1:28:16
That you have to log that you have to log because here it won't log automatically.

Tirth 1:28:17
So it knows what.
No, but when we get the response back, when we're getting the answer back, we can update the chat history inside the same state.

Tarun Jain 1:28:33
Then you'll have to define that attribute here.

Tirth 1:28:36
Yes, that is what I'm saying. Like we can have the chat history over here with list of dictionary question and answer, so query and answer.

Tarun Jain 1:28:42
Uh.
Now that is possible, yeah.

Tirth 1:28:47
Yeah. OK. Got it. Got it, I think. OK.
But then we will also have the previous context with it.

Tarun Jain 1:28:57
But we'll have to ensure that you keep it buffer memory or certain window. If it reaches Pi, you have to keep removing the old ones.

Tirth 1:29:06
Right, right, right. I, uh, Hardip gave me a good option. Like what they do is they summarize the last 10 chats and then they give it as a context.

Tarun Jain 1:29:15
You want that is there. So if you just search for memory and langchain.
Uh, so if there is somewhere summary.

Hardip Patel 1:29:30
OK.
Yes.

Tarun Jain 1:29:34
So if you see this one summarizer, this one conversation summary buffer memory.

Tirth 1:29:34
Somebody, somebody.

Hardip Patel 1:29:36
Yeah.

Tirth 1:29:37
Mm.
Mhm.

Tarun Jain 1:29:40
You can use this one.

Tirth 1:29:42
I see. Okay.

Tarun Jain 1:29:46
So this is mainly used for long term memory. What will typically happen is let's suppose you have 10 previous experience but only three are relevant. So how do you know which are relevant? You use search. So if you want to use search, you use vector database.

Tirth 1:29:47
OK.

Hardip Patel 1:29:57
OK.

Tirth 1:30:00
Um.

Tarun Jain 1:30:03
So this is for memory.

Tirth 1:30:03
Mm.
That is also true. Yeah, OK.

Tarun Jain 1:30:06
It will take too much of time if you use summary.


Tirth 1:30:10
Mm.

Tarun Jain 1:30:11
But if you are using something for internal team purpose then fine.


Tirth 1:30:15
OK.

Tarun Jain 1:30:17
Is this done the chatbot?


Tirth 1:30:19
Yeah.


Hardip Patel 1:30:20
Mm.

Tarun Jain 1:30:20
So what is the command?


Hardip Patel 1:30:25
Stream it run.


Tirth 1:30:25
Run app PY.

Tarun Jain 1:30:29
Streamlet run.

Hardip Patel 1:30:32
Yes.

Tirth 1:30:32
I have.

Tarun Jain 1:30:33
I have no see why.
You'll get a local host URL.

Tirth 1:30:39
No module name line graph. I missed something.

Tarun Jain 1:30:44
Yeah.
Here I'll just ask what does.

Hardip Patel 1:30:53
Yeah.

Tirth 1:30:57
It says no module name Langra for me.

Tarun Jain 1:31:00
Can you check if you are inside environment variable? If you see this is generating response, this is the spinner. Can you see this?

Hardip Patel 1:31:06
Mm.
Yeah, yeah.

Tarun Jain 1:31:09
Now how did I get this? By using it inside if condition.
So if you see as soon as I click on button, what is the button get response have with

stream rate dot spinner generating response. Can anyone tell me where we saw with before?
Where did we use with earlier?

Hardip Patel 1:31:34
Read.

Tirth 1:31:35
I remember I mistook it for a while, but it was with with when.

Tarun Jain 1:31:40
Do you remember when we used with?

Tirth 1:31:43
Uh.

Tarun Jain 1:31:49
We haven't looked many examples full with, but we used it for three or four four examples we used.

Hardip Patel 1:31:56
OK.

Tarun Jain 1:31:58
You remember these three keywords?

Hardip Patel 1:32:02
5 append.

Tarun Jain 1:32:05
Uh, we used.
We should use with a lot. I guess we need to have anything related to files because when we use pandas we didn't use with because we directly have functions called read CSV. Even for line chain we directly have loaders we didn't use with.

Hardip Patel 1:32:23
Um.

Tarun Jain 1:32:28
But internally all this functions uses with so with St. dot spinner whatever message you want to give. This is the fallback message, then colon, then your final response.

Tirth 1:32:42
I'm still getting this error no module name langraph.

Tarun Jain 1:32:47
Are you inside environment variable?

Hardip Patel 1:32:48
Yeah.

Tirth 1:32:50
I am with her source ENVSI have dot ENV with me.

Hardip Patel 1:32:52
So.

Tarun Jain 1:32:54
Do one thing open in terminal and then run it there.
But do you think this is a good chatbot here? Every time you have to remove this and you have to create it again, right?

Hardip Patel 1:33:05
Sorry.
Um.

Tirth 1:33:09
But.

Tarun Jain 1:33:10
So stream data also provides chat bot based interface. There is a template code for it. I'll just give the template code.

Hardip Patel 1:33:29
I mean.

Tarun Jain 1:33:31
How many of you are able to run this on streamlet?

Hardip Patel 1:33:34
Not yet. I didn't try.

Tirth 1:33:38
Able to run it but.

Tarun Jain 1:33:40
But you guys tried with this one, right? You do is dot PY. It is very simple, just St. dot whatever you need.

Hardip Patel 1:33:43
It.
Yeah.

Tarun Jain 1:33:47
And if you get confused, just check for the particular thing. Click on metric. The documentation is well written for every single component with examples.

Hardip Patel 1:33:57
Stop.

Tarun Jain 1:34:02
And let me paste that boiler code that I have.

Tirth 1:34:07
It's asking St. dot session state has no key widget this. Did you forgot to initialize it?

Tarun Jain 1:34:13
Can you share your screen?

Tirth 1:34:19
I'm.

Tarun Jain 1:34:21
We're still here, everyone have copied. If yes, then probably I'll proceed with the the actual chat bot where you have it like ChatGPT.

Tirth 1:34:31
So.

Hardip Patel 1:34:31
Mhm.

Tarun Jain 1:34:32
Just cross check if you are.

Tirth 1:34:32
We have this I'm I'm inside SRC and then I have app prompt and utils. Very similar. Same thing that we did. It all worked with this one. Now I have main with all these details.

Hardip Patel 1:34:46
OK.

Tarun Jain 1:34:49
Once.

Hardip Patel 1:34:49
Your your screen is not uh.

Tirth 1:34:52
Oh, it's not visible yet.

Hardip Patel 1:34:53
Yeah.

Tirth 1:34:55
Thank you.
Is it visible now?

Hardip Patel 1:35:07
Yeah, yes.

Tarun Jain 1:35:08
Can you do a list?
Can you do a list once?

Tirth 1:35:12
Elis.

Tarun Jain 1:35:13
I mean, where are here?

Tirth 1:35:15
Yeah.
So I'm here with SRCVNVN requirements and it is VNV as selected and inside LS hyphen AN SRC.

Tarun Jain 1:35:30
Why is it this?
It's starting with base I guess.

Tirth 1:35:33
Sorry.
Beef.

Tarun Jain 1:35:38
Base base.

Hardip Patel 1:35:41
Not best.

Tarun Jain 1:35:44
So if you see that it's based before, is it from Condor?

Hardip Patel 1:35:49
In maybe, oh, this.

Tarun Jain 1:35:50
No, uh.

Tirth 1:35:50
This.

Tarun Jain 1:35:52
Before teeth. So your name is correct.

Tirth 1:35:54
Before did.

Tarun Jain 1:35:58
Above above that there is a base.

Tirth 1:36:00
Oh, this one.

Because I just did LS. Uh, this is. So anything I think would just go with that it it has nothing to do with.

Hardip Patel 1:36:06
Yeah, this one's this.
OK.

Tarun Jain 1:36:10
OK, OK.

Tirth 1:36:12
Yeah, it's, uh, it's the terminal.

Tarun Jain 1:36:13
Extreme lit run SRC out of viewer.

Tirth 1:36:17
Same late run app SRC app dot PY. It's throwing this error.
But again, that is another thing. I started it from Cursor. It was working fine, but it was throwing a different error from Cursor terminal.
So you used it.

Hardip Patel 1:36:34
Oh, then you you have to do the source.

Tirth 1:36:39
I did the source. I'll I'll do it again. Uh, VENV.

Hardip Patel 1:36:43
Yeah, you.
Then.

Tirth 1:36:46
So I did this and if I do this it is there but it comes with error.

Tarun Jain 1:36:47
Bin.

Hardip Patel 1:36:50
And yeah, now it will run.

Tirth 1:36:58
So if I do uh.
What is?
What the heck? OK, it's happening.

Hardip Patel 1:37:10
OK, it's the worst.

Tirth 1:37:17
Every time it will do this.

Tarun Jain 1:37:18
OK.
M.

Hardip Patel 1:37:21
I don't think so, I guess.

Tarun Jain 1:37:21
No, no, that is loading right.

Tirth 1:37:24
I don't know. OK, I think it is working here. I I might have missed something. Thank you.

Tarun Jain 1:37:29
OK, so let's there is one final part which is left on how to make the chatbot better. I'll share one template code.

How do I pay this?
I'm not able to paste anything here.

Hardip Patel 1:38:05
No, no.
No, no, no. Can you try to copy it again and paste?
Yeah.
No.

Tarun Jain 1:38:20
Why not?

Hardip Patel 1:38:33
Sometimes they have to raise for a message.
Yeah.

Tarun Jain 1:38:41
Oh, what the hell?
OK, so just copy this template code.

Hardip Patel 1:38:45
So.
Yeah.

Tarun Jain 1:38:48
And if you just search for Streamlit chatbot right, you'll find the same code.

Hardip Patel 1:38:49
Good.

Tarun Jain 1:38:54
Chatbot design.
So if you see, you'll have WhatsApp. I will just say hi.

Tirth 1:39:06
So we remove, we remove the main.

Hardip Patel 1:39:12
Oh.

Tirth 1:39:12
OK.

Tarun Jain 1:39:13
No, we can add it with main as well. So if you see it will be like this now.
Are you able to see the screen?

Hardip Patel 1:39:18
But.

Tirth 1:39:19
Yes.

Hardip Patel 1:39:20
Yeah.

Tarun Jain 1:39:20
So this is the basic LLM LLM chatbot and the code is same where I've exposed the code.
This is the one that's the chatbot.
Hi.

Hardip Patel 1:39:36
Maybe not.

Tarun Jain 1:39:42
Where did I get that code from?
SD messages. Can you just check if this is there this line?

Hardip Patel 1:39:50
Um, yeah.

Tarun Jain 1:39:52
What you guys can do is you can keep this code, copy this code and I will just verify where I got that.
So this is the function lines and then you have messages role and content. So now if you see here whatever you have here St. dot chat message then messages role. This is not from LLM.

Hardip Patel 1:40:00
OK.

Tirth 1:40:12
Mm.

Tarun Jain 1:40:12
This is coming from the chat message of the object which is created inside string data chat message.
Is this clear?

Hardip Patel 1:40:20
OK.

Tirth 1:40:21
OK.

Tarun Jain 1:40:22
And now I'll come back here.
And.
Where is that code? I'll paste it here.
And this has to go inside main function.
So graph is there. Can you remove this line? Can you come back to the try?

Tirth 1:40:44
Mhm.

Tarun Jain 1:40:46
So here graph, just keep it graph. Remove this filter condition.
And instead of thinking, just make it generating response.
Can you make these two changes?
One is generating response instead of thinking and then remove filter condition.
And then copy this entire thing. Messages should be outside.

Hardip Patel 1:41:14
OK.

Tarun Jain 1:41:14
This thing is the red outside.
These two things, I'll keep it outside.
And then what are you supposed to remove from query? Remove everything or you can just comment.
For comment, make sure you use.
Oh.
Hash time. So what will happen is if you keep triple quotes, it will add that particular information on the screen. So that's the reason why I'm using.

Hardip Patel 1:41:38
Edge.

Tarun Jain 1:41:50
What is this hashtag?
Can you confirm if you have made these changes?
So whatever code I shared in the chat, just pick the if messages outside main function because this is managing state inside main function wherever you're defining this query where we just used chat text input.

Tirth 1:42:05

Mhm.

Hardip Patel 1:42:06

But.

Tirth 1:42:12

Mm-hmm.

Tarun Jain 1:42:12

So what is this next input doing? It is creating a entry box for the user to enter any text. Once you have that query, you have a button and then you're passing it to the text. So this is not an ideal way to have a chat bot right? Every time you have to enter a message, then enter a button and you only see one message at.

Hardip Patel 1:42:25

It.

Tarun Jain 1:42:31

At the time, if you need multiple messages after graph, just copy this entire code.
And we have to use.
Total two this thing we have to enter. One is 1 tab.
And then another tab.
This is stop.

Hardip Patel 1:43:06

Yeah.

Tarun Jain 1:43:09

Yeah.

Hardip Patel 1:43:10

M.

Tarun Jain 1:43:12
Let me just check if I made it correctly.
I'll send you the main function. Let me just verify this once.
OK, there is one if condition which is wrong.
They should come in.

Hardip Patel 1:43:48
OK.
So I I then complete a Uh code, but I just removed the last query and added Uh code.

Tarun Jain 1:44:12
Which query?

Hardip Patel 1:44:13
And no, whatever we were doing manually, I mean the response and all I removed and I just use this after workflow compile and it.

Tarun Jain 1:44:30
Can you?

Hardip Patel 1:44:31
Streams will be working, sorry.

Tarun Jain 1:44:33
Can you share your screen?

Hardip Patel 1:44:35
Yeah.
How can I share it? I am getting the output.

Tarun Jain 1:44:39
Are you getting the output?

Yeah, you know I'm getting, but the only thing is the initial message is they should not show.

Hardip Patel 1:44:44
OK.
Oh, sorry, let me check.

Tarun Jain 1:44:54
So now if you see this human thing is there, right? That is human message and then you will have the LLM response.

Hardip Patel 1:45:01
OK, OK. Oh yeah, it does show.

Tarun Jain 1:45:08
Somewhere up and down we have made some mistake.

Hardip Patel 1:45:09
OK.

Tarun Jain 1:45:15
If you see this should not come here, it should come below.

Tirth 1:45:20
Hmm.

Tarun Jain 1:45:21
And this I have to fix it.
Meanwhile, just copy this. Let me try to fix this error.

Hardip Patel 1:45:32
Sorry, I don't understand. I think, uh, let me share this thing.

Tarun Jain 1:45:33
But.

Hardip Patel 1:45:42
Because I don't share.

Tarun Jain 1:45:49
If I remove the main function, it's working properly.

Hardip Patel 1:45:56
Yeah.
I am also using filter. I haven't removed it. Is it visible?

Tarun Jain 1:46:01
I
Oh yeah, can you ask a new prompt?

Hardip Patel 1:46:08
Yeah, what is the purpose in design the for content and error?

Tarun Jain 1:46:20
Yeah, for you it's working. How are you defining it? Can you come back to the code?

Hardip Patel 1:46:25
Yeah.

Tarun Jain 1:46:26
Is it inside? OK, you're not, I mean.

Hardip Patel 1:46:26
So.
I just it after directly after this.

Tarun Jain 1:46:32
Yeah, this is working fine. Even in my case, if I don't define it inside main, it's working fine. If you see here, there is no main in my screen.

Hardip Patel 1:46:33
Uh.
OK.

Tarun Jain 1:46:42
Oh, can you copy this one or everyone?

Hardip Patel 1:46:44
This one.
Uh uh oh, I base this.

Tarun Jain 1:46:49
No, I've pasted it already. Even I'm not using main. Now this works fine.

Hardip Patel 1:46:51
OK.
OK, OK.
I think, uh, you need to share this thing.

Tarun Jain 1:47:04
Yeah, one second.
Oh, is my screen visible now?

Hardip Patel 1:47:16
Not yet, yeah.

Tarun Jain 1:47:16
It.
OK, so you understood what these two lines is, right? This is title and this is.
Sub bidding and then you have chat bot interface. Here you can ask an e-mail.
What is the what's the location of the office?

Hardip Patel 1:47:38
Yes.

Tarun Jain 1:47:49
So here instead of invoke we have to use streaming so that we can have one word at a time, 1 token at a time.

Hardip Patel 1:47:50
OK.
8.
Complete my alarm.
Yeah.

Tarun Jain 1:48:05
Taking time.

Hardip Patel 1:48:07
Yeah, actually.
M.
Mm.

Tarun Jain 1:48:36
Now if I ask anything, what is the location of the office?
That clear chat should be on sidebar so that I can remove it whenever I need instead of being it on the home screen.

Hardip Patel 1:48:50
Which implement like and Python you have to delete for all the OK.

Tarun Jain 1:48:58
So you just have to do streamlit dot sidebar dot button and here you have clear button. You can also close this if you don't want.

Hardip Patel 1:49:00
M.
Hmm.

Tarun Jain 1:49:06
Then settings.
You can use dark theme.

Hardip Patel 1:49:10
Yeah.

Tarun Jain 1:49:13
And if you want to deploy, you can directly use dot streamlit dot IO.

Hardip Patel 1:49:13
No, I'm not.

Tarun Jain 1:49:20
Here what you're supposed to do is you have to create app.
Whatever, uh, this thing is there, right? Just push it to GitHub.
And then you have to add that repo URL. Then branch will typically be main. Let me show one of the stream data.
So you'll have main, then this will be app dot PY and you can have any name here.
Now click on advanced settings. What you're supposed to do is instead of these two lines, can you see this dot ENV?

Hardip Patel 1:49:58
Mm.

Tarun Jain 1:49:59
Remove this dot ENV and remove this line.
And here what you're supposed to do is you have to do OS dot environment Google API key then streamlet dot secrets.

Hardip Patel 1:50:10
M.

Tarun Jain 1:50:15

You have tried Google API key streamlet dot secrets. You need to have quadrant URL.
Streamlit dot secrets.

Hardip Patel 1:50:30
I'm in the.

Tarun Jain 1:50:32
Quadrant API key. Just make this three.

Hardip Patel 1:50:34
Oh.
Tarun, this also works for me. OS environment also works for me.

Tarun Jain 1:50:42
No, this is much better because this will keep, uh, more safe.

Hardip Patel 1:50:45
Oh.
Mm.

Tarun Jain 1:50:49
So just copy this three things.

Hardip Patel 1:50:49
You don't.

Tarun Jain 1:50:54
This is the syntax.
Is this clear guys? So you have to convert or get T in secrets and whatever key you use here, same case you have to define here but with the codes.

Hardip Patel 1:51:00
Yeah.

Tarun Jain 1:51:12

So that format you can click on this format you will get it. So if you see key equals then value it should be inside quotes.

And once this is done, click on Save.

It is under advanced settings and then click on deploy. The only two things you need is app dot PY and requirements dot txt. Since we are using prompts also you can also upload that app dot PY prompts dot PY requirements dot txt.

And then change this as secrets and then you can deploy the app by keeping whatever you need.

So you can try deploying it and let me know if you were able to deploy or not.

This can be 1 task and obviously we will improve the speed. We should not have this much of slow application and one thing is we'll replace this embeddings when we deploy it and instead of invoke we will use stream. Where is the invoke?

Here if you see instead of invoke will use stream and I hope everyone remembers this stream stream logic. So this will generate generator. Instead of right we have something called a stream date dot text stream.

Hardip Patel 1:52:22

Mm.

Yeah.

Tarun Jain 1:52:36

User function text stream.

Right stream.

Tirth 1:52:42

But then we won't have the result answer right, Avina?

Tarun Jain 1:52:49

Uh.

There will be some delay, but still it will be faster than invoke, at least.

Tirth 1:52:55

No, no, I'm, I'm talking about line #97. We are getting the answer of response over here that that won't be right. We won't have that right.

Hardip Patel 1:53:04
OK.

Tarun Jain 1:53:05
Huh. So here what you can do is you can just add results, result, then answer it.

Tirth 1:53:15
But sure.

Tarun Jain 1:53:15
So here if you see this is a generator object. Earlier what we did was we used to use for.
Chunk end result.

Tirth 1:53:26
Hmm.

Tarun Jain 1:53:28
Then use use St. dot write.
You know.

Hardip Patel 1:53:33
Mhm.

Tarun Jain 1:53:37
Just print chunk.

Hardip Patel 1:53:42
So.
Um.

Tarun Jain 1:53:47
I hate this auto complete.

Tirth 1:53:49
Hey, the portal.

Tarun Jain 1:53:51
So this was the syntax.
So this will return a generator object. This generator object needs to iterate over every single chunk and this is how we used to define. But in streamlet you have a function called right stream. So this right stream will take the iterator. Can you see the data type?

Hardip Patel 1:54:07
Yeah.
1.

Tirth 1:54:12
Yes.

Tarun Jain 1:54:13
Iterator. So this is also iterators.

Hardip Patel 1:54:15
I mean.

Tirth 1:54:24
OK.

Tarun Jain 1:54:26
So here they've given a dummy text example and then they're streaming the data.
But here this is already a iteratable object so we can directly use stream.
Let me know if you're able to do this. If not, I'll show you one example.

But I hope you understood how Langraph works and stream it very simple. Stream it is just use St. then whatever object you need.

Tirth 1:55:00
M.

Tarun Jain 1:55:03
There are too many visits.
And if you want to use anything in HTML, what you can do is define markdown and here you can have your HTML code and then there is one function called unsafe allow HTML to be true.

Hardip Patel 1:55:08
What?

Tirth 1:55:20
It says generator object is not subscriptable.

Hardip Patel 1:55:21
Yes.

Tirth 1:55:28
With the bacon chips, yeah, with this answer.

Tarun Jain 1:55:28
Oh, are you using answer? Are you?

Hardip Patel 1:55:31
After.

Tarun Jain 1:55:32
So.

Tirth 1:55:35

I think we might just have to change it to right.
Stream.

Tarun Jain 1:55:40
Someone already deployed the app. Are they?

Hardip Patel 1:55:42
Hmm.
Not the line graph one.

Tarun Jain 1:55:51
Oh, which one is this?

Hardip Patel 1:55:53
Uh, this is using Lang Jain, the leveling guide.
OK.
It will say I don't know anything.
OK.

Tirth 1:56:19
Yeah, I think you have to. You do have to give.
Which position? I don't know what it will give back.

Hardip Patel 1:56:28
Well, it was. I don't know, I guess.

Tarun Jain 1:56:33
I don't. If it takes this much time, I don't think it will generate. I don't know.

Tirth 1:56:33
Really.

Hardip Patel 1:56:39
OK.

Tarun Jain 1:56:40
So I don't know in the sense it will be be less than.

Hardip Patel 1:56:43
I mean like if he does not have the knowledge of this then.

Tirth 1:56:46
OK.

Hardip Patel 1:56:49
It was. It is a fallback, no that that knowledge does not have this.

Tarun Jain 1:56:54
But is this right information? Is it from the document? This is too lengthy.

Tirth 1:57:00
It is. It is. The document is very lengthy. You have to trust on this one 300 page documents around 3 MB of file.

Tarun Jain 1:57:08
Response correct that we have to verify or is it generic because it's a generic question.

Tirth 1:57:12
These things, it is a very generic question. It is not related to any particular.

Tarun Jain 1:57:18
OK.

Hardip Patel 1:57:19
Yeah, but like I if I get top K then I'm returning or else I will get.

Tirth 1:57:19
Position.

Tarun Jain 1:57:25
OK, I don't think KPIs and all will be given in generic.

Hardip Patel 1:57:31
What?

Tirth 1:57:32
No, no, we have KPIs in the doc as well.

Tarun Jain 1:57:35
Uh, then fine, this is fine if I run this on LGPT.

Hardip Patel 1:57:39
Yeah.

Tarun Jain 1:57:58
So.
OK, this is different key activities and checklist. Then again puts.

Hardip Patel 1:58:04
Mm.
Hmm.

Tarun Jain 1:58:08
Uliya.
But we'll have to improve the speeding process. This is very slow. Even what I did was very slow, which is not ideal. We'll improve the speed.

Hardip Patel 1:58:15
Mhm.

Tirth 1:58:18
OK.

Hardip Patel 1:58:20
OK.

Tarun Jain 1:58:21
The only issue with it is the state management, so.

Tirth 1:58:40
When was it faster?

Tarun Jain 1:58:41
That will be fast only.

Hardip Patel 1:58:43
Yeah, little bit fast.

Tarun Jain 1:58:45
Because that is API based and it's running on their servers.

Hardip Patel 1:58:48
Yeah.

Tarun Jain 1:58:54
And also it's better to use Azure because we'll get too much of free credits.
And way too much free credit.

Hardip Patel 1:59:00
Yeah, we don't.

Tirth 1:59:01
We registered for Azure. I got the $1000 but then it is asking me for my registration information which I don't have.
Anyways.

Tarun Jain 1:59:13
Uh, you got into founder startup or?

Tirth 1:59:14
No, this is good. This is good too.
With another domain name, atentic.com is already registered with Microsoft now, so they are not allowing us to be on founders hub.

Hardip Patel 1:59:22
Good.
OK.

Tarun Jain 1:59:26
Oh, OK.

Hardip Patel 1:59:27
I don't know.

Tarun Jain 1:59:30
Kulia, this is it. Tomorrow we'll try to have our own vector database custom creation with filtering technique so where you can also add metadata. So far we have never used metadata.

Hardip Patel 1:59:42
OK, question in the text.

Tirth 1:59:44
Hmm.

Tarun Jain 1:59:47
Oh, what?

Tirth 1:59:50
That's good.

Tarun Jain 1:59:53
You can play around with stream rate. It's really very amazing. So when we do EDA, stream rate is best for EDA where you can see too much of graphs and all.
So for report generation, you can just deploy streamlet. Hey, this is our CSV file. This is report we generated and we usually use to send stream data. This was earlier times when we.

Tirth 2:00:13
Hmm.

Tarun Jain 2:00:15
But now if we deploy and send the LLM based application, it's been misused, properly misused.

Hardip Patel 2:00:22
2.

Tirth 2:00:24
Sure.

Tarun Jain 2:00:25
So it is only better when you show demo. If you deploy send it to customer they'll they'll always have bad feedback only.

Tirth 2:00:28
Mm.
Hmm.

Tarun Jain 2:00:34
So that is one thing you'll have to keep in mind that avoid sharing screen lit application app URL to anyone.

Hardip Patel 2:00:40
Yes.

Tarun Jain 2:00:43
Unless you first show the demo and then share it.


Tirth 2:00:43
OK.

Tarun Jain 2:00:49
OK. Uh, any questions you have in streamed it and uh Langra?


Tirth 2:00:52
No good so far.

Tarun Jain 2:00:55
Thanks.


Hardip Patel 2:00:55
We need to practice more for this.


Tirth 2:00:58
Yeah, but then you could like.

Tarun Jain 2:00:59
Practice. I'll be very honest. Stream lit. You just have to know one line import streamlit dot St.
Then it's only documentation.


Hardip Patel 2:01:09
Mhm.

Tarun Jain 2:01:12
And that documentation is also just St. dot whatever you need and then brackets you'll have documentation. Specifically you don't even have to open documentation on different tab, it is within the code only.

Hardip Patel 2:01:23
Yeah, good.

Tarun Jain 2:01:27
But language you need practice.

Hardip Patel 2:01:28
For me.
****.

Tirth 2:01:32
Mhm.

Tarun Jain 2:01:34
Oh, cool. Uh, that's it, right?

Hardip Patel 2:01:36
What?

Tarun Jain 2:01:37
Uh, there was only two assignment which was submitted on the Tableau one. Did anyone try it?

Tirth 2:01:38
Thank you.

Hardip Patel 2:01:46
I tried the Tavali.

Tarun Jain 2:01:50
I believe it.

Hardip Patel 2:01:51
Sharma.

Tarun Jain 2:01:52
Also Tabli and uh, Google Switch. Sorry, not Google Switch. Finance.


Tirth 2:01:53
And uh, yeah, Yahoo, Yahoo minutes.


Hardip Patel 2:01:54
OK.
Yeah, and then I just replaced it.

Tarun Jain 2:02:00
Cool. OK, but that is fine. That app, I mean that code is very dummy code, but when we do the project right from 12th to 17th, that will be very useful.
Because we'll not have sessions as well, right? So you you will have to have some continuity. So during that time you can work on those projects.


Hardip Patel 2:02:23
OK.

Tarun Jain 2:02:23
Cool, yeah. Thanks.


Tirth 2:02:24
OK. Thank you. Thank you, Tarun.


Hardip Patel 2:02:26
Thank you. Thank you.

2:02:27
OK.

Ajay Patel stopped transcription