But I just want to break the system down, so do you understand all of the moving pieces and if if you were a total beginner, you'll understand this a little little bit better better after I break it down so we start with the user and the user is going to send in a message and ultimately this assistant right here is going to send a message. In fact the user and they're going to carry out the conversation, but the real magic is in what happens in the middle of the system here the first place that's going to go to a memory search. This memory is going to take with the user so let's imagine that the user says what is my name it's going to send that query up to a database of memories and this is called a vector database. This is basically just all of the embedding of all the information that you have organized in a way that AI can understand it. Example query would be what is my name or username and it's going to send that into the database database is going to respond semantically by giving relevant memories that are related to that users inquiry right here once any memories are pulled down from this. They're going to have a lot of metadata with them extra stuff, necessarily things like the file ID where this was found, but the number of memories that were found what vector store is actually in all of this stuff that isn't necessarily required format, kind as to just pull out the memory data explaining that memory so that this assistant can understand that context going into this user what is my name to the assistant assistant isn't going to know out of the box what your name is but if that's stored in the memory database then this assistant is going to be able to get that information from this system right here what makes this system self learning well the assistant has two options they can either just respond to the user or they can also add new memories to the memory database user that isn't database right here is able to actually add that memory so if they say my name is Carter, it goes through queries. It doesn't find any information in the vector database here when he gets to the assistant the assistant has the option to add that as a memory of it never forgets it creates somewhat of a flywheel where the memories in the vector database are getting stronger every time that you go through and have a conversation with this assistant and it's never going to forget those TV details and preferences that you played actually built the system out and I'm going to show you guys how to build it as well open eyes agent builder and what I have here is a memories vector different files that hold a bit of information so we have this one right here this one here and this one right here so right now we have these three memories so now I'm just going to go ahead and test it out. We have a fresh conversation here and I'm just going to say what is my name and as you can see here it's going to conduct the file search as a guy how it's going to transform that and now the assistant is responding and it says your name is so I got it. Perfect I got it right my name is Carter because that is something that stored in my vector data let's say that I wanted to add new memory to my database so that whenever I come back to this conversation, this agent is going to remember that I can just say something like I'm currently filming a tutorial on how to use OpenAI new agent builder in order to create self learning agent agents I can send this off. The file is going to do it thing. It's going to transform all of my existing memories, and then here is going to reason through whether or not it should upload that to the vector store as you can see here it wants to upload to vector store and this is where it's actually in memory memories. Eyes suggesting they may create educational content about AI tools in the future it's going to be able to query this memory with the virtual so I'm just going to hit approved to save that memory and now I'm just getting atypical normal response from the assistant so all of that kind of happens in the background and all of these steps here can be hidden from the end user I'm only showing you this in the development preview right now

now let's close this chat and reopen a new one and I'm just going to say what do you know about me and I'll send that off file search tool is going to work it transformed those and now the assistance is coming back with the irrelevant content. Here are the memories that returned your name is Carter you have a dog named Reese you're filming a tutorial on how to use opening eyes agent builder to create self learning agents and you prefer short eloquent to the point responses. These are the four memories that the file search tool was able to pull back in to go back into my memories, vector store here as you can see it has all four of these memories stored as finals in today's I'm going to show you exactly how to build anti deploy that cell learning agent, but to be honest if you wanted to learn how to do this open eyes documentation is a bit scattered. This is a new tool and if you wanted to find exactly how to build whatever you want to build with agent builder online, there's not really many places to do that so whatever I recommend is, if you're interested in really mastering, this agent builder thing so that you can deploy production grade agents to your clients or for your company you should jump into our agent builder master course in agent builder mastery we're going to start with the basics of mastering nose these little rectangles on the screen inside of agent builder that act as the building blocks for your agent. We really break this down so that it's simple for beginners and we give you guys all kinds of chapters that are going to help you through whatever it is that you're struggling with how to build a couple advanced agent examples and finally I'm going to walk you through how to deploy your AI agents and I'll give you all the employment requirements and then we'll go ahead and deploy the agent with this tool that I don't for you guys that allows you to deploy it in a corner chat model like this or on the screen of your website that your users can interact with it so if you want to access this course while it's still 50% off, you can head over AI foundation.io and you can check out our workshop here. You can purchase a course if you'd like it and you can get some more information and then you can make a decision on whether you want to join there now I'm going to start with the basics. We're going to develop this agent architecture step-by-step and then I'm going to show you. How did the to get to opening eyes agent builder you're going to head over to platform.openingeye.com and then then you're gonna head over to the dashboard link up here you're gonna head over to the dashboard link up here that's open up the screen screen and then you can go to agent builder on the left-hand side and you can click workflow and it's going to give you a basic workflow to start with. I'm just going to delete my agent to begin with and we're going to add in a file search tool and I'm just going to connect that to start getting all of our notes out here and then we'll go ahead and configure that in a moment next thing we need is a transformer and then we need to connect our assistance drag agents to assistant, so I can rename these by adding a name on whichever option click click on the agent. I can just call this guy assistant. Let's click on file search here and I'm going to create a store storage inside memories check to confirm that I'm going to copy we go builder perfect. The next thing I can do is I can show how many results that I want to bring in for the query we're going to add context and input as what this does is it just pulls the text from the previous start note right here so that outputs this input is text and then the file search is going to use that to search this vector database so if I user over here and I said what is my name then it's going to use that as the query for the database it's going to use some AI tools help us grab that information. It's going to pull back anything relevant related to what is my name as the query but if I preview this and I just run it really quick I can say what is my name as you can see it's going to hit all of the tools, but the workflow is going to fail when he gets the assistant

because it's not pulling anything in from the transform if I want to monitor, what's going on with these test runs that I'm doing I can always click on the evaluate button in the upper right hand corner and that's going to show me it's going to show me where we ran into an error. This workflow called transformer so let's see what happened just before that. We used the file search. I'll click into the file search and I'll see if that I have the input and I don't have any results being output because that is an empty vector store so what we can do for now is going to transform here and for the value I can click into here and I can see the schema that that's getting returned. It can return all of this information and for now just to keep this simple. Let's just click on the text string just so that it pulls back the zero index memory. This means the first memory that it sees using zero index zero also stands for the first that's how computer programming works so now if if I run this, this should work on the transform step so I can say what is my name it still isn't gonna have my name, but it should get through the transform step and click click again as you can see it still didn't work so let's just do input results. Let's just get everything that falls through. I'll just do the results and we'll try this again. What is my name that time and the assistant actually ran and gave me a response so that's perfect but now we need this assistant to be able to actually upload memories so if I go into the assistant, you're going to notice is it has this tools right here by the tools there isn't a way right now to actually upload to the file search, I only have file search, which is doing what this says right here but just read the files that are in there, but how can we upload to that file search? Upload to custom build this MCP server and I'm not going to go to in-depth on how I built it in this video but I am going to have a resources link below that goes along with this entire video so that you guys can kind of slow it down I'm even going to have a copy button that you can just copy the entire post and you can paste it into your favorite AI tool to help you along the way I want to run my help. Just run it over here locally I've already developed it so I can just send this right here and you can see it's now a server and point which is basically just the public way to show whatever my computer computer is running locally on a certain court. This is up and running and it just needs a few things so we're going to do first. We're going to need the environment variable my opening eye APP and since I have that saved here in the code for this, I can actually just grab that environment variable for the open AI keep going to my assistant here and I can click the button and hit MCP server and then I can add in a server I could put in my token down here and then for that URL I'm just going to type in whatever my MCP servers are out/SSE and then memory uploader tool connect as you could see. I have this tool here called upload to vector store and this is all running on my server so let's go ahead and add this tool now let's just give us some basic instructions. I can just say you are a helpful assistant and you receive any memories that are already relevant to the conversation here and then I can give it any of the memories that came from the transformed note so if I just add contacts, I can add the result because remember we just have the result coming out here so I add the result here we go and then I could say you can also upload these memories if you have any new memories to add, you can also upload new memories if the user expresses any preference that's not in the previous memories given above make sure to always upload the users memory preferences, using your upload memory tool that you have access to and then we can just insert the users actual message here it's already going to receive a message if we have include, but if I just add in the user message here that will just make it a lot easier for this assistant to to understand the difference between any previous memories that it has up here versus what the users actual raw

messages so I'll go ahead and add in the context now let's go ahead and test this out. I'm just going to hit the start button right here and then I'll go ahead and send a message saying what is my name if I sent this off the first time it should not know my name because we haven't told my name in the past and it's not in our memories so it says I don't know your name if I say my name is Carter and I sent this. It should hold me to the inspector store upload approval so it's just asking if I want to approve this new memory. The username is I'm going to go ahead and approve and now it should remember that my name is Carter now let's go go ahead and close this and go back into a fresh chat just to test this out so I'll go ahead and send a message again. What is my name and if I send it this time it should be able to pull back what my name is as you can see it says your name is Carter and the reason that it's able to do that is because in my store it is now saved into tutorial memories. A new file right here in this file represents the my name is memory. This is going to work just fine, but if I go to the transform, I can actually get rid of. I'll have this link and the description below with the resources so that you guys can use this as well and what is it just checks if we have memories at each level of the index and it adds up to 10 memories in here so that is the max that we have set on the file search 10 max results and if you wanted to, you could just add more cases if you want to allow more maximum memory stay pulled into the context now let's go ahead and deploy this agent because a lot of people have trouble deploying these opening eye agent build their workflows to the live Internet so what you can do is you can just hit publish in the upper right hand Quarter and publish again and now that it's published, it's going to give me a workflow ID so to get that workflow ID I'll just click on the code right here and then I see workflow ID we're about to get into deployment. This is the exciting part but it can be a bit complex, especially if you aren't a developer if you haven't had experience as a programmer in the past, so if you want these step-by-step, hold your hand all the way through nitty-gritty of how to actually go through the development and deployment process and I recommend checking out our agent builder master course link will be in the description below to make deployment easier. I have a starting server. It's just called easy server and this just makes it a whole lot simpler to deployed your jacket server so I can actually just clone this and then I can just change a few of the variables and I can get this running out of my local device so I'll just see to documents and then I'll see my projects and then I'll make a directory for tutorial deployment and then I'll see into that next. I'm just going to have and get clone and then I'm going to paste that URL it's just hub.com/barcodes/easy-check-server.the enter and now it has clone to that repository so if I just copy the name of it now I should be able to see the into that and now we're in there. OK so if I open this up in BS code, I should be able to see that we have an example environment variables here so we have the opening AIAPIT we have the work full ID with the domain base CRL and environment head over to.studio/playground and it's going to bring you to this chat builders so you can build out this chat widget however you want to see it on your website or wherever you're deploying this agent too I'm just going to make a couple changes. I pretty much like how I have this, but I do want to change the surface color here so I can just change the color of the background make it like a mint color. Maybe I'm gonna go with a mint color for our little button down here as you can see and then once I'm done editing this, I can go into my checking options and copy the code there head to our easy chat server I've got a code or any code if you want you can just paste your configuration here. This is just the code you get from. I just want it into confit.TS and we're ready to deploy this thing so I just close this and save it the easiest way that I deploy this is to just

upload this to get help so I can get help and creating new repository. I'm just going to serve server. I'm going to make sure that the visibility is on private go ahead and deposit now I'll copy this link right here while I'm in this folder. I'm just going to commit this to get up to manually. I also have a course in our classroom if you're interested in document into that development process just say something along the lines of I want to upload this to here that we're going to be using. You can also manually do this, but this is a bit quicker. It's just going to push what we have here to get and as you can see, it said our code has been pushed to this upper depository so I copy this URL here and then I head over to that in my web browser I should see my code then upload it here. Perfect now I'm going to head over to railway.com and I'm just going to create a new new project and I'm going to select to get hundred repository then we'll go ahead and configure the hub app and I'm just gonna go down and I'm going to add in that repository so I can just search for it right here and just type in my and I should see my check server and save now if I refresh, I should see it come in right here my server so I'll go ahead and click on it and this should deploy directly from the hub to a website that they've been up for the first time that you run this it's going to crash but if you go into your variables, you can go ahead and enter your environment variables here like you're opening a VIP your jacket works full ID or domain public key base URL you can also go to your settings here and if you scroll down, you can generate a domain for your server and I'm just going to have this on or that's fine. I'll go ahead and generate the domain and I can copy the assuming and then in my variables I'm gonna add that as the base URL and I can just give it HTPS at the beginning then after you fill out all of your environment variables, you can read deploy and unemployment. I can monitor the deployment and then after the deployment is successful, you can just go to the website you generated and once you're here, you should be able to see that you have this review of car widget if you click on it, it should open up the chat in your basement created so as you can see here it has the green accent to be added and if I say what is my name it's able to go through the process silently need to find what my name is and it also brings back some other context as well, and if I wanted to amend this on a website website I can just copy the embed code, but if I wanted this on a public facing website, I give you something like lovable, or I could embed it on a WordPress website anywhere that I want on the Internet and what I can do here as I can just say make a simple page with this and bed on it and I can paste in the script that my server gives me it's getting to work now creating that website website then if I preview it out of the box as you can see it's not working, but if I went ahead and published this and just hit publish now it's going to give me this domain here that I can copy. I'll add this to my. I'll add this to my domains and open AI and save changes then if I actually head over to this website, I should see that it loads out perfectly right here so now anybody that goes to this website can utilize this chat in her face and we can also have a chat history and it's going to construct that for them right here. I hope you enjoy today's guide and if you wanna learn more about opening eyes agent builders, and check out our course agent master, let me know when the comments what you guys thought of this and what you need more help with when it comes to using agent builder I'll see you guys in the next one.

Yes, an OpenAI Agent Builder agent can follow those steps, but only if they are converted into clear, structured instructions and workflow nodes, and even then it will follow them probabilistically, not perfectly.

How agents follow steps

- The model reads your instructions (system prompt, node prompts) and tries to execute them, but it is not a deterministic script runner, so vague or very long step lists are more likely to be skipped or mis-ordered.
- Reliability improves when you:
- Break the process into separate nodes (Start → File Search → Transform → Assistant → MCP tool, etc.) instead of one giant text block.
- Make each step a concrete action ("call this tool with X and Y", "map resultsi.text into a list", "only upload memory if…").

What to do with your step-by-step

- Use your step-by-step as a design spec, not as one big prompt dump. Turn each main step into:
- a node in the Agent Builder workflow, or
- a short, focused instruction block tied to that node (e.g., a Transform prompt, an Assistant system message, a tool description).
- Then test in the Agent Builder preview: run multiple conversations and see where it deviates, and tighten instructions or add guardrail nodes if it skips or reorders steps.