**AI MakerSpace Notion Week 2**

<aside>

**🎬 Live Session Resources, recorded September 16th**

🖼️ [Slides](https://www.canva.com/design/DAGzJw-3i34/1UdGr5HlXlPjFtabOAWdkw/edit?utm_content=DAGzJw-3i34&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton) & 🎥 [Recording](https://us02web.zoom.us/rec/share/7UJErmXFPnBQmIxWoeHVtCVcjtF1c_XmzAybJLGgei5Xrju_Q2jgPzgjYI8YT06o.pRQgg0m-t4-HHAmV)(\*6zDd0%S)

🧑‍💻 [GitHub Repo](https://github.com/AI-Maker-Space/AIE8/tree/main/03_End-to-End_RAG)

📝 [Homework](https://forms.gle/ZVvwkbg4jEpHKpCY9)

🗣️ [Feedback](https://forms.gle/9SnYW7vgNLeGpkh47)

</aside>  
<https://www.notion.so/Session-3-End-to-End-AI-Applications-2025-Industry-Use-Cases-26acd547af3d80b4b646e2fd6f1fd31c>

**Overview**

Today, we will bring everything that we’ve discussed so far in the course together into one session, then we will layer in some updated information about industry use cases and challenge you all to take your builds even further with open source (a.k.a. OSS, or Open-Source Software) Large Language Models.

Technically, there is **no pre-work** for today’s class. However, if you have not yet set up your dev environment or Git workflow properly, you will run into issues. Further, if you want to take your build to the next level today with remotely-hosted OSS LLMs, you’ll have the option to do just that!

<aside> 🌟

The guiding question is simple: *can you take Session 2’s Pythonic RAG application and combine it with* [*The AI Engineer Challenge*](https://aimakerspace.io/aie-challenge/) *E2E LLM application?*

</aside>

This will be the first E2E AI application prototype - the first of many throughout the cohort!

We will keep the stack streamlined and simple, although we will detail the use case. Importantly, this application will get you out of the notebook and working in version-controlled application development environments aligned with state-of-the-art LLM, RAG, and agentic application development.

🧱 The full stack

* **LLM:** [OpenAI GPT-4.1-mini](https://platform.openai.com/docs/models)
* **Embedding Model:** [OpenAI text-3-embedding-small](https://platform.openai.com/docs/guides/embeddings/what-are-embeddings#:~:text=*%2D-,ada,-%2D*%2D001)
* **Orchestration**: [OpenAI Python SDK](https://github.com/openai/openai-python)
* **Vector Database:** Custom, Pythonic
* **User Interface:** Vibe-Coded
* **Deployment:** [Docker](https://www.docker.com/), [Vercel](https://vercel.com/)

<https://github.com/a16z-infra/llm-app-stack>

We need to begin thinking *end-to-end*, both from a use-case perspective and from an infrastructure point of view. To this … end … we will also discuss emerging industry standards for creating business value with production LLM applications.

**📛 Required Tooling & Account Setup**

* You’ll need all the required tooling and account setup from [Session 1](https://www.notion.so/Session-1-Introduction-and-Vibe-Check-263cd547af3d81869041ccc46523f1ec?pvs=21) and [Session 2](https://www.notion.so/Session-2-Context-Engineering-I-Retrieval-Augmented-Generation-RAG-26acd547af3d8041a75bfa162d1ab600?pvs=21)!
* [Optional] For today’s 🚧 Advanced Build, you will need to set up an API Key for Together AI.

[api.together.xyz](https://api.together.xyz/sso-signin?redirectUrl=%2Ffine-tuning)

**🧑‍💻 Recommended Pre-Work!**

1. Review [The AI Engineering Bootcamp Challenge](https://aimakerspace.io/aie-challenge/) and ask yourself “do I understand what each of the End-to-End components are in this stack?”
2. Review your Homework 2 on Pythonic RAG and ask yourself “do I understand what components were added to the LLM application stack?” Note that *in Session 2, we only added backend components*.
3. 🤔 Spend ***time thinking*** about what you should be building and why for Demo Day! Start by reviewing [OpenAI’s Six Use Case Primitives](https://cdn.openai.com/business-guides-and-resources/identifying-and-scaling-ai-use-cases.pdf) (Content creation, Automation, Research, Coding, Data analysis, Ideation / strategy) and come up with one idea for at least two categories. Use the GPTs we’ve been introduced to for help!
   1. <https://chatgpt.com/g/g-h5aUtVu0G-chatgpt-use-cases-for-work?model=gpt-4o>
   2. <https://chatgpt.com/g/g-UwSQ1o8AW-build-ship-share-gpt>
4. 🤓 Check out additional reading material in [Go Deeper](https://www.notion.so/Session-3-End-to-End-AI-Applications-2025-Industry-Use-Cases-26acd547af3d80b4b646e2fd6f1fd31c?pvs=21)

**🕴️ Industry Use Cases**

During Session 3, we will go deep on a single use case that we’ll use to build additional assignments on. To best align with the way that the industry is creating huge value today, we will align our primary cohort use case with one of the most popular multi-agent systems that every model provider/lab has built into their offering: Deep Research.

However, we also want to go deeper to understand the landscape of applications and use cases across the industry today. To this end, it’s important to understand what folks are working on on out at the LLM edge and what they’re hearing from their customers and clients.

***Bottom line: In 2025, people expect ROI from AI, and it’s important to deliver it.***

The key word of the year is probably productivity.

Is your use case something that creates additional productivity through time savings? It’s probably a good year to build it.

Not sure what use case you should work on? Well, you can use the most recent usage data from [OpenAI](https://openai.com/index/how-people-are-using-chatgpt/) and [Anthropic](https://www.anthropic.com/research/anthropic-economic-index-september-2025-report) to help you find out.

**Big Picture**

According to [HBR](https://hbr.org/2025/04/how-people-are-really-using-gen-ai-in-2025) (April 2025), top GenAI use cases shifted from Generating Ideas to Therapy/Companionship! Interestingly, we also see that “Specific Search” has been downgraded from the top use cases on the list.

This is interesting. Why? Because we can see from LangChain’s 2024 [State of AI Agents](https://www.langchain.com/stateofaiagents) that the number 1 use case for agents was **research and summarization**. That is, searching and researching. That is, enhancing search and retrieval processes. In other words, fancy RAG; a.k.a. [Deep Research](https://www.notion.so/Session-3-End-to-End-AI-Applications-2025-Industry-Use-Cases-26acd547af3d80b4b646e2fd6f1fd31c?pvs=21); our cohort’s primary use case. This capability to search and re-search has been built into all of the leading LLM products. From ChatGPT to Gemini, Claude, Grok, and more, deep research capabilities became table stakes for LLM providers to give their customers.

Agents has, of course, been the the buzzword of 2025 (and perhaps ‘agentic systems’ as well). As we’ll learn, Deep Research systems *are multi-agent systems*. It’s important to understand that when you use ChatGPT, it’s no longer just a model, but rather [a complex multi-agent application](https://www.youtube.com/live/OkqnAk1eH4M?si=J4UgHQcHg-ofa4fH).

Moreover, as [KPMG put it](https://kpmg.com/us/en/media/news/q2-ai-pulse-2025-agents-move-beyond-experimentation.html) in June of 2025: “AI-agent strategies are moving past the experimentation phase,” and that “AI-agents transition from tools to teammates.” They also noted that “ROI focus shifts amid data challenges and need for board expertise.” *This means that ROI and scoping business problems worth solving is more important than ever*, and a far cry from May 2024 when McKinsey broke the business-value issue down in one sentence: “[Gen AI starts to generate value](https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai).”

Over the past few years, an interesting pattern has emerged: the top consulting use cases become a feature in the apps built by leading LLM model providers (and now leading agent labs).

While from 2024 to 2025 we saw it play out with Report Generation into Deep Research tooling, we might also point out that from 2023 to 2024 we saw a similar evolution with respect to to code generation, which continues to be an absolutely killer use case for emerging productions/companies in Silicon Valley, including [Cursor](https://cursor.com/en)/[Anysphere](https://anysphere.inc/) and [Devin](https://devin.ai/)/[Cognition](https://cognition.ai/) (acquirers of the [Windsurf](https://cognition.ai/blog/windsurf) team), and others. It’s worth mentioning that [ChatGPT Codex](https://openai.com/index/introducing-upgrades-to-codex/) and [Claude Code](https://claude.com/product/claude-code) fit into this hierarchy of different levels of [Coding Agents](https://www.youtube.com/live/S0EwhDf94JI?si=vUchjUL_NDoRrwuO&t=644) as well.

So this begs the question - **what are the top use cases of 2025 so far that we should expect to see become killer products in 2026?**

**Use-Case Resources from Industry Leaders**

Luckily, we have reports dropped in September of 2025 to help us anticipate what is coming, including [How People Use ChatGPT](https://www.nber.org/papers/w34255) and the [Anthropic Economic Index report: Uneven geographic and enterprise AI adoption](https://www.anthropic.com/research/anthropic-economic-index-september-2025-report). Looking across both of these reports, we can see an interesting trend clearly stated by the anthropic team with evidence from OpenAI that appears to support the same (or at least a similar) conclusion: ***Educational and scientific tasks continue their rise in relative importance*.**

**Figure 1.1:** [**Claude.ai**](http://Claude.ai) **usage over time.** Each panel shows the share of sampled conversations on [Claude.ai](http://Claude.ai) associated with tasks from each SOC major group. We see notable increases in usage for scientific and educational tasks. SOC major groups ranked by usage in our first report. [[Ref](https://www.anthropic.com/research/anthropic-economic-index-september-2025-report#:~:text=Educational%20and%20scientific%20tasks%20continue%20their%20rise%20in%20relative%20importance)]

Figure 9: Breakdown of granular conversation topic shares within the coarse mapping defined in Table 3. The underlying classifier prompt is available in Appendix A. Each bin reports a percentage of the total population. Shares are calculated from a sample of approximately 1.1 million sampled conversations from May 15, 2024 through June 26, 2025. Observations are reweighted to reflect total message volumes on a given day. Sampling details available in Section 3. [[Ref](https://www.nber.org/system/files/working_papers/w34255/w34255.pdf)]

*What conclusions do you draw from this recent usage data?*

Beyond the latest off-the-press data drops, we can consider wider perspectives and learnings from top AI companies (references below) and ask what is the best way to go about identifying use cases that deliver real ROI?

Answer:

<aside> 📈

The highest-return AI projects in 2025 start small, ship quickly, and measure ruthlessly. Work closely with teams through ideation workshops and/or hackathons to rank ideas **with a “value versus feasibility” lens, using evaluation appropriately to be able to drive iterative development quickly and within six months.**

Leadership provides cover; domain experts provide traction; metrics keep everyone honest. Follow that loop and, per Google’s benchmark, you should see tangible ROI in the first half-year—and have the data to prove it.

</aside>

Context:

* [Identifying and Scaling AI Use Cases](https://cdn.openai.com/business-guides-and-resources/identifying-and-scaling-ai-use-cases.pdf), by OpenAI (April, 2025)
  + ***Frame the hunt****:* teach staff six “use-case primitives”👇. Run ideation workshops and hack-days so every team maps recurring friction points to one of those primitives.

| **#** | **Primitive** | **What it means (in plain English)** | **Typical high-ROI tasks** |
| --- | --- | --- | --- |
| 1 | **Content creation** | Producing or rewriting text, images, audio or video at speed and to spec. | Drafting emails, ads, blog posts; repurposing decks; localising content. |
| 2 | **Automation** | Triggering actions or completing routine steps end-to-end once a condition is met. | Filling forms, generating invoices, routing support tickets, updating CRM. |
| 3 | **Research** | Gathering, searching, summarising or comparing information faster than a human alone. | Market scans, competitive intelligence digests, literature reviews, Q&A chatbots. |
| 4 | **Coding** | Writing, refactoring, documenting or testing software and scripts. | Generating boiler-plate code, unit-tests, SQL queries, API stubs, infrastructure as code. |
| 5 | **Data analysis** | Inspecting structured/unstructured data to find patterns, calculate metrics or create visuals. | Exploratory data analysis, forecasting, segment clustering, dashboard commentary. |
| 6 | **Ideation / strategy** | Sparking and structuring new concepts, decision frameworks or plans. | Brainstorming product features, campaign themes, go-to-market plans, OKR drafts. |

* [AI In the Enterprise](https://cdn.openai.com/business-guides-and-resources/ai-in-the-enterprise.pdf), by OpenAI (May, 2025)
* [Generative AI Adoption Index](https://press.aboutamazon.com/aws/2025/5/generative-ai-adoption-index), by AWS (May, 2025)
* <https://cloud.google.com/resources/roi-of-generative-ai>
* <https://hai.stanford.edu/ai-index>
* <https://www.stateof.ai/>

In the end, we can summarize much of this in a single word to help you as AI Engineers and leaders trying to further your career: ROI can be equated toproductivity these days. Our experience shows us this, and much of the data we mention in today’s session sheet either implicitly or explicitly appears to support this. Perhaps in 2026, we can not just be more productive when writing code and doing research code and research, but perhaps also teach and learn?

In other words, ss will become important for your demo day pitch to consider, *how much time are you saving with AI?*

**🧑‍🔬 Deep Research**

The killer app of 2025; the multi-agent system that helps us search and research, has seen broad adoption throughout the industry, and will serve as our primary cohort use case that will build from scratch using OSS tools.

Doing D**eep Research**; that is, autonomously exploring, gathering, and synthesizing information from various sources (e.g., search tools, reference documents, or code execution), is the kind of thing that if you can build in an intelligent way for your domain and your organization, you will be indispensable to leadership looking to undergo an AI Transformation.

Let’s consider how many organization have releasing their own “Deep Research” agents. The common goal here is to produce a system that is capable of:

* Constructing a research plan.
* Searching or browsing the web.
* Summarizing, analyzing, and refining large volumes of information.
* Presenting findings in a clear and comprehensive format.

Here is a quick breakdown of releases we’ve seen since late 2024

December 11, 2024: [Google’s Deep Research](https://blog.google/products/gemini/google-gemini-deep-research/)

“Deep Research uses AI to **explore complex topics on your behalf** and provide you with findings in a comprehensive, easy-to-read report”

February 2, 2025: [OpenAI’s Deep Research](https://openai.com/index/introducing-deep-research/)

“Deep research is OpenAI’s next agent that can **do work for you independently** - you give it a prompt, and ChatGPT will **find, analyze, and synthesize hundreds of online sources** to **create a comprehensive report at the level of the research analyst**.”

**July 17, 2025 update:** Deep research can now go **even deeper and broader** with access to a visual browser as part of ChatGPT agent. To access these updated capabilities, simply select “agent mode” from the dropdown in the composer and enter your query directly. The original deep research functionality remains available via the “deep research” option in the tools menu.

February 4, 2025 (just 2 days later!): [Hugging Face Open Deep Research](https://huggingface.co/blog/open-deep-research)

While powerful LLMs are now freely available in open-source (see e.g. [**the recent DeepSeek R1 model**](https://huggingface.co/deepseek-ai/DeepSeek-R1)), OpenAI didn’t disclose much about the agentic framework underlying Deep Research…

So we decided to embark on a 24-hour mission to reproduce their results and open-source the needed framework along the way!

The clock is ticking, let’s go! ⏱️

February 14, 2025: [Perplexity Deep Research](https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research)

Deep Research takes question answering to the next level by spending 2-4 minutes doing the work it would take a human expert many hours to perform. Here’s how it works:

* **Research with reasoning** - Equipped with search and coding capabilities, Perplexity’s Deep Research mode iteratively searches, reads documents, and reasons about what to do next, refining its research plan as it learns more about the subject areas. This is similar to how a human might research a new topic, refining one’s understanding throughout the process.
* **Report writing** - Once the source materials have been fully evaluated, the agent then synthesizes all the research into a clear and comprehensive report.
* **Export & Share** - You can then export the final report to a PDF or document, or convert it into a Perplexity Page and share it with colleagues or friends.

February 19, 2025: [Grok 3 Beta - The Age of Reasoning Agents](https://x.ai/news/grok-3)

**Grok Agents: Combining Reasoning and Tool Use**

To understand the universe, we must interface Grok with the world. Equipped with code interpreters and internet access, Grok 3 models learn to query for missing context, dynamically adjust their approach, and improve their reasoning based on feedback.

As a first step towards this vision, we are rolling out DeepSearch—our first agent. It's a lightning-fast AI agent built to relentlessly seek the truth across the entire corpus of human knowledge. DeepSearch is designed to synthesize key information, reason about conflicting facts and opinions, and distill clarity from complexity. Whether you need to access the latest real-time news, seek advice about your social woes, or conduct in-depth scientific research, DeepSearch will take you far beyond a browser search. Its final summary trace results in a concise and comprehensive report, to help you keep up with a world that never slows down.

April 15, 2025: [Claude Research](https://www.anthropic.com/news/research)

**Research**

Research transforms how Claude finds and reasons with information. Claude operates agentically, conducting multiple searches that build on each other while determining exactly what to investigate next. It explores different angles of your question automatically and works through open questions systematically. This approach delivers thorough answers, complete with easy-to-check citations so you can trust Claude's findings.

Research delivers high-quality, comprehensive answers in minutes, making it practical for the multiple research tasks you tackle throughout your workday. It's this balance of speed and quality that sets it apart.

July 17, 2025: [Mistral AI Deep Research](https://mistral.ai/news/le-chat-dives-deep?utm_source=alphasignal&utm_campaign=2025-07-21&asuniq=bd20930c)

**Dive deeper with Deep Research**

Research mode turns Le Chat into a coordinated research assistant that can plan, clarify your needs, search, and synthesize. Ask a meaty question, and it will break it down, gather credible sources, and build a structured, reference-backed report that’s easy to follow.

It’s powered by our tool-augmented Deep Research agent (in preview), but designed to feel simple, transparent, and genuinely helpful, as though you’re collaborating with a well-organized research partner.

<aside> 🧪

Of course, using AI to accelerate research is not a new idea. This is something that many scientists have been interested in for a long time. There are breakthroughs happening all the time that are tangentially related to doing “Deeper Research.” For instance, consider [AlphaEvolve](https://deepmind.google/discover/blog/alphaevolve-a-gemini-powered-coding-agent-for-designing-advanced-algorithms/) and their [math breakthrough](https://www.youtube.com/watch?si=Xp8pHr-RjWb6YOUY&v=vC9nAosXrJw&feature=youtu.be) just a few months ago.

</aside>

We look forward to building our own Deep Research application throughout your cohort!

**[Bonus] 🤩 AI Makerspace Demo Day Use Cases**

One competitive advantage we have at AI Makerspace to help you decide what to build and why is that we’ve been doing Demo Days with our students since July 2023 (and at other companies before that!)

It’s instructive to look across use cases that our cohorts have been actively working on in their companies while taking the Bootcamp. As always is true when you’re working on the edge, it’s best to start with the most recent data about building, shipping, and sharing.

Check out Cohort 7’s projects directly [here](https://www.youtube.com/playlist?list=PLrSHiQgy4VjH8PIgmZ-I_VbKt8aKWwLG_) 👇, and it might even be of particular interest to attend to the projects that are not live on YouTube and the creators opted to keep secret for one consideration or another. You can find those by comparing the [Demo Day Run of Show](https://www.notion.so/255cd547af3d8009904ac61e20f9d14a?pvs=21) to the Cohort playlist. To check out all the projects presented in AIE7, check out the full [Rehearsal Run of Show](https://www.notion.so/256cd547af3d80289c3cdbdd1de49a9f?pvs=21).

[The AI Engineering Bootcamp, Cohort 7 Demo Day!](https://www.youtube.com/playlist?list=PLrSHiQgy4VjH8PIgmZ-I_VbKt8aKWwLG_)

Some of our favorite end-to-end project prototype examples past Demo Days are highlighted in our new [Best-of The AI Engineering Bootcamp Demo Day](https://www.youtube.com/playlist?list=PLrSHiQgy4VjEVMLYVDYmtY8Gu8JitS39g)” playlist!

Take note in particular of [HTM-LLM](https://youtu.be/yXtW9XwXIAY?si=Zuka0EKh3rMrpLH6) from the VA Medical Center (learn more about their work [here]([https://aimakerspace.io/gen-ai-upskilling-for-teams/#:~:text=prem hardware](https://aimakerspace.io/gen-ai-upskilling-for-teams/#:~:text=prem%20hardware)....-,VAMC,-The%20Department%20of)) and of the work done by Hrvatski Telecom on [Report Wiz](https://youtu.be/7-XJjZDKESU?si=CmjfRHoObqFO8y3A) and [3N1](https://youtu.be/9dzLWXO7VWI?si=7jWfHVRYBI-0lQS6) in Croatia (learn more [here](https://aimakerspace.io/gen-ai-upskilling-for-teams/#:~:text=Day%3A%20HTM%2DLLM-,Hrvatski%20Telekom,-The%20Croatian%20division)).

Also, [Garret from DeepWriter](https://x.com/DeepAIWriter) (demo day [presentation](https://www.youtube.com/watch?v=-KtyQDSKF0s)) was in one of our original cohorts, and this guy [sold his code for $10k](https://www.youtube.com/watch?v=rWjjfQVztpg) within a few weeks of the cohort!

And of course, don’t forget about Cohort 4 legends [Publicus](https://publicus.ai/) and [aithon](https://aithon.ai/login) that are still running strong companies today!

**🤩 ROI is for You and/or a Company**

It’s important to understand what your ROI is for this project.

Perhaps you’re not working for a company, or not particularly thrilled about the use cases in the company that you are working for.

In this case, we highly recommend that you **don’t do anything you’re not excited about**!

It will be much more meaningful, valuable, useful, and a bigger return on your investment to build something for a community that you’re involved in - even if it’s just a simple chatbot that pulls data from facebook posts and that users that you hang out with regularly in your personal life can derive some enjoyment from!

We’ve built the [Build🏗-Ship🚢-Share 🚀 GPT](https://chatgpt.com/g/g-UwSQ1o8AW-build-ship-share-gpt) to be your personal ideation coach to help you come up with:

1. Ideas associated with your industry.
2. Ideas for a community you’re a part of.
3. Ideas that combine the two.

Don’t forget - the focus right now, when it come to deciding on your project, is to figure out:

* **Problem**: Write a 1 sentence description of the problem you're trying to solve
* **Why**: Write a paragraph about why this is a problem worth solving. To the company, to the world, to you!
* **Success**: Write 1-2 sentences about what success looks like. What is the Key Performance Indicator?
* **Audience**: Specify the user/stakeholder/customer that this product is for. What is their pain point or need that connects back to the problem?
* **Potential Solution**: Briefly describe data that you might use and the modeling techniques that you plan to leverage.
* **Sharing**: What community (or company) do you plan on sharing your project in?

**🕳️ Go Deeper!**

* 🤖 For today’s advanced build, we suggest leveraging the open-source [gpt-oss](https://openai.com/index/introducing-gpt-oss/) model (August, 2025) that we did a deep dive on [here](https://www.youtube.com/live/nyb3TnUkwE8?si=Yz97mSOltslcPnIM).
* 📜 We saw recent drops from the leading model providers that provide deep insights into how people are using their products. Both are worth checking out in greater detail

[How people are using ChatGPT](https://openai.com/index/how-people-are-using-chatgpt/)

[Anthropic Economic Index report: Uneven geographic and enterprise AI adoption](https://www.anthropic.com/research/anthropic-economic-index-september-2025-report)

# 📜 As AI and Makerspace nerds, we also love following the grand visions for embodied AI we are also starting to see (e.g., for ROBOTS!). Our favorite is from [Jim Fan of Nvidia](https://youtu.be/_2NijXqBESI?si=KtCOh2OdLuNM81Sv). **Session 4: RAG with LangGraph, OSS Local Models, & Eval w/ LangSmith**

<https://www.notion.so/Session-4-RAG-with-LangGraph-OSS-Local-Models-Eval-w-LangSmith-26acd547af3d80838d5beba464d7e701>   
<aside>

<aside>

**🎬 Live Session Resources, recorded September 18th**

🖼️ [Slides](https://www.canva.com/design/DAGzMO1y0FQ/oJaw4HMIFecP3oX9jSO4fw/edit?utm_content=DAGzMO1y0FQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton) & 🎥 [Recording](https://us02web.zoom.us/rec/share/jEs9TS_re1f9X3y2T61Dgv_bEp6EmVzVkiYDOC-cEU8WA2tR5jMI1bwsn4L_Al1n.msDqlCRCROFBaRCH)(78y?PRTg)

🧑‍💻 [GitHub Repo](https://github.com/AI-Maker-Space/AIE8/tree/main/04_Production_RAG)

📝 [Homework](https://forms.gle/i2SdxgWX4ahFwNrCA)

🗣️ [Feedback](https://forms.gle/ymYqK5MBLAG11jDB9)

</aside>

**Overview**

In Session 4, we dig into LangChain and LangGraph! LangChain has emerged as the leader of the pack of infrastructure [orchestration tools](https://github.com/a16z-infra/llm-app-stack?tab=readme-ov-file#orchestrators).

LangChain is a framework for developing applications powered by large language models (LLMs). [[Ref](https://python.langchain.com/docs/introduction/)]

LangChain, as a company, has built an entire ecosystem of tools (we’ll often refer to this as the Lang-X ecosystem).

During this session, we’ll learn about LangChain, LangGraph, and LangSmith. LangChain and LangGraph are used for orchestration (we can think of LangGraph as the next evolution of LangChain, well-suited for agents and agentic reasoning loops).

LangGraph is built for developers who want to build powerful, adaptable AI agents. [[Ref](https://langchain-ai.github.io/langgraph/concepts/why-langgraph/)]

LangSmith, on the other hand, is used for monitoring, visibility, logging, and evaluation.

**LangSmith** is a platform for building production-grade LLM applications. [[Ref](https://docs.smith.langchain.com/)]

During Session 4, we will also cover a specific use case that we’ll build on in the next session. From prototype to production with LangChain, let’s gooooooo!

**📛 Required Tooling & Account Setup**

1. Create an ollama account

[Sign in to Ollama](https://ollama.com/signin)

1. [Optional] Create a LangSmith account 👇

[LangSmith](https://smith.langchain.com/)

**🧑‍💻 Recommended Pre-Work!**

In this class, we’ll learn each of the key constructs of LangChain, and we’ll build and evaluate a RAG application with LangChain. Here’s what to do to prepare:

1. 🏫 During this session, we’ll begin exploration of our cohort’s use case, inspired by the many Deep Research applications that every model provider has rolled out in 2025!
2. The purpose of LangGraph (and subsequently, [LangGraph Platform](https://langchain-ai.github.io/langgraph/concepts/langgraph_platform/)) becomes apparent the more you study the idea of state machines. As an introduction, check out CEO Harrison Chase’s TED Talk on Cognitive architectures.

[The magical AI assistants of the future — and the engineering behind them](https://www.ted.com/talks/harrison_chase_the_magical_ai_assistants_of_the_future_and_the_engineering_behind_them?subtitle=en)

1. 🤓 Check out additional reading material in [Go Deeper](https://www.notion.so/Session-4-RAG-with-LangGraph-OSS-Local-Models-Eval-w-LangSmith-26acd547af3d80838d5beba464d7e701?pvs=21).

**⛓️ Core Constructs: LangChain**

When orchestrating complex LLM applications that leverage context and reasoning, LangChain has emerged as a best-practice tool. To begin, it’s instructive to start with how we can leverage the pattern of Retrieval Augmented Generation to build apps using LangChain.

What are the core components we need to understand?

Let us start, where we should, with the abstraction of the chain in LangChain.

<aside> ⛓️ Chain: a sequence of calls to other components

</aside>

With this idea of chains in mind, we can understand that LangChain Expression Language, or LCEL, allows us to ***compose chains*** easily. LCEL also enables us to build production-grade prototypes that can be deployed with no code changes.

**A Simple Chain - Chat Models and Prompts**

The first chain we’ll build with LCEL combines Model and Prompt from Models I/O.

Using LLM models that have been instruction-tuned and fine-tuned for chat is a best practice for LLM-powered applications. Just as we learned about the {System, User, Assistant} roles in OpenAI, we can leverage chat-style models using the {System, Human, AI} roles in LangChain. These are, for all intents and purposes, the same.

Chat completions roles, OpenAI vs. LangChain.

\*\*\* Recall that with o1 models and newer, developer messages replace previous systemmessages.\*

Building our first LLM chain requires a chat prompt template. We have seen this in our Pythonic RAG system, and can see many other examples [here](https://python.langchain.com/docs/how_to/#prompt-templates). Here is what our first chain looks like in code.

chain = chat\_prompt | openai\_chat\_model

**Our Second Chain - Retrieval Augmented Generation**

To do RAG with LangChain, we leverage the same process we used to build our Pythonic RAG application. Below these steps are outlined with the relevant LangChain constructs and documentation that we’ll use during Session 4!

1. Create **Database**
   1. Document Loader [[Ref](https://python.langchain.com/docs/how_to/#document-loaders)]
   2. Text Splitter [[Ref](https://python.langchain.com/docs/how_to/#text-splitters)]
   3. Embedding Model [[Ref](https://python.langchain.com/docs/integrations/text_embedding/)]
   4. Vector Store [[Ref](https://python.langchain.com/docs/how_to/#vector-stores)]
2. Ask **Question**
   1. Embedding Model [[Ref](https://python.langchain.com/docs/integrations/text_embedding/)]
3. Find **References**
   1. Retriever [[Ref](https://python.langchain.com/docs/how_to/#retrievers)]
4. **Augment** the Prompt
   1. Prompt [[Ref](https://python.langchain.com/docs/how_to/#prompt-templates)]
5. **Generate** a better answer!
   1. Model [[Ref](https://python.langchain.com/docs/integrations/chat/)]

<aside> ⛓️ In the end, we want to create a single RAG chain that leverages LCEL

</aside>

Here’s an example in code of such a RAG chain👇

simple\_rag = (

{"context": retriever, "query": RunnablePassthrough()}

| chat\_prompt

| openai\_chat\_model

| StrOutputParser()

)

Notice that we see the word “Runnable” in the code.

**The Runnable**

In LangChain, a Runnable is like a LEGO brick in your AI application - it's a standardized component that can be easily connected with other components. The real power of Runnables comes from their ability to be combined in flexible ways using LCEL (LangChain Expression Language).

*Every component of a chain is a* ***runnable****.*

In fact, **the primary abstraction in the LangChain ecosystem is the runnable**.

We often refer to a Runnable created using LCEL as a "chain" [[Ref](https://python.langchain.com/docs/concepts/lcel/)]

**Key Features of Runnables**

**1. Universal Interface**

Every Runnable in LangChain follows the same pattern:

* Takes an input
* Performs some operation
* Returns an output

This consistency means you can treat different components (like models, retrievers, or parsers) in the same way.

**2. Built-in Parallelization**

Runnables come with methods for handling multiple inputs efficiently:

# Process inputs in parallel, maintain order

results = chain.batch([input1, input2, input3])

# Process inputs as they complete

for result in chain.batch\_as\_completed([input1, input2, input3]):

print(result)

**3. Streaming Support**

Perfect for responsive applications:

# Stream outputs as they're generated

for chunk in chain.stream({"query": "Tell me a story"}):

print(chunk, end="", flush=True)

**4. Easy Composition**

The | operator makes building pipelines intuitive, as seen above! e.g.;

# Create a basic RAG chain

rag\_chain = retriever | prompt | model | output\_parser

**Common Types of Runnables**

* Language Models: Like our ChatOpenAI instance
* Prompt Templates: Format inputs consistently
* Retrievers: Get relevant context from a vector store
* Output Parsers: Structure model outputs
* LangGraph Nodes: Individual components in our graph

Think of Runnables as the building blocks of your LLM application. Just like how you can combine LEGO bricks in countless ways, you can mix and match Runnables to create increasingly sophisticated applications!

**🕸️  Core Constructs: LangGraph**

Instead of building our entire RAG chain using runnables, the best-practice in 2025 is to use LangGraph directly.

<aside> 🔄 LangGraph lets us **add cycles** to applications built on LangChain.

</aside>

The essence of LangGraph is that it uses graphs to add cycles.

**Why Cycles?**

We can think of a cycle in our graph as a more robust and customizable loop. It allows us to keep our application ***agent-forward*** while still giving the powerful functionality of traditional loops.

Due to the inclusion of cycles over loops, we can also compose rather complex flows through our graph in a much more readable and natural fashion. Effectively allowing us to recreate application flowcharts in code in an almost 1-to-1 fashion.

**Why LangGraph?**

During this session, *we will be using LangGraph as a Directed Acyclic Graph (DAG*). Beyond the agent-forward approach - we can easily compose and combine traditional DAG chains with powerful cyclic behavior due to the tight integration with LCEL.

In this way, LangGraph is a natural extension to LangChain's core offerings!

**Graphs**

Graphs are collections of connected objects: nodes and edges.

* **Node**: Think function or runnable; i.e. *something that changes* ***state***
* **Edge**: Think path to take; i.e., *where to pass* ***state*** *object next*

A state object is initially defined by passing a state definition to a class representing the graph. This state object, or StateGraph, gets updated over time. The agent's internal state is represented simply as a list of messages. Remember how we interacted with the OpenAI API with a list of messages with roles? Same idea.

Just as every component of a chain is a runnable, each node in our graph can be a runnable, or even an entire chain!

Welcome to the next layer of abstraction.

**⚒️ LangSmith for Evaluation & Visibility**

**LangSmith** is a platform for building production-grade LLM applications. [[Ref](https://docs.smith.langchain.com/)]

In addition to building out a RAG application with LangChain, we will also learn what the developer sees behind the scenes when users interact with our LangChain application! Enter … LangSmith.

“LangSmith turns LLM “magic” into enterprise-ready applications.”

Ultimately, we will use LangSmith for evaluation and for visibility.

**Evaluation**

The key to understanding the right way to do quantitative evaluation with tools like LangSmith is Metrics Driven Development (MDD). There are three steps to keep in mind:

1. Establish a baseline
2. **Change stuff** that potentially improves baseline
3. **Recalculate** metrics

It’s not about absolute values. Rather, MDD is about *relative changes* in evaluation metrics.

We will explore LangSmith further in future sessions!

**🕳️ Go Deeper!**

* 🤖 For today’s primary build, we’ll be leveraging the open-source [gpt-oss](https://openai.com/index/introducing-gpt-oss/) model (August, 2025) that we did a deep dive on [here](https://www.youtube.com/live/nyb3TnUkwE8?si=Yz97mSOltslcPnIM). We will also be leveraging the newly-released [EmbeddingGemma](https://developers.googleblog.com/en/introducing-embeddinggemma/) model from Google (September, 2025).
* <https://blog.langchain.com/the-rise-of-context-engineering/> (June 2025)
* [Is LangGraph used in Production?](https://blog.langchain.com/is-langgraph-used-in-production/) (Feb 2025)