

Homework — The LLM & Agent Era

Estimated time: ~45 min · No code you have to run · A browser, a free account on a couple of AI products, and a notebook

Three tool-tour exercises that put you in front of the three shapes of LLM product you'll keep meeting in 2026: a *chat assistant* (and the differences between the big three), a *retrieval-augmented* reader that grounds answers in your own documents, and an *agent* that actually carries out a multi-step task. By the end you'll have felt — not just read about — the knowledge-cutoff problem, what citations are for, and the gap between a chatbot reply and an agent run.

Before you start

Keep a one-page notebook open (paper or a text file). For each task, write the short answers under (a)/(b)/(c). You are not turning this in — the goal is to build the muscle of *poking at* an LLM product instead of just reading about one. There is **no code you need to run** in this homework. Free accounts on the products mentioned are sufficient as of 2026; this is a fast-moving field, so if a UI element has moved by the time you read this, pick the nearest equivalent.

The three tasks

Task 1 — Side-by-side: ask three chatbots the same questions

Open three browser tabs: claude.ai, chatgpt.com, and gemini.google.com. Sign in with free accounts — all three vendors offer a free tier as of 2026 (a paid tier exists too, but the free tier is enough for this exercise).

Ask **the same three questions** to each of the three chatbots. Use the same wording in all three tabs so the only variable is the model:

1. *"In two paragraphs, explain what an LLM is to a non-technical friend."*
2. *"What were the most important AI news stories in the last 30 days? Be specific."*
3. *"What is your knowledge cutoff date?"*

Take a screenshot of each Q1 answer so you can compare them afterwards.
(You're not grading them — you're feeling the style differences.)

Self-check — answer all three:

- **(a)** Whose explanation in Q1 felt clearest to *you*? There is no wrong answer — note the difference in style (length, use of analogy, presence of headings, tone). Three frontier LLMs reading the same prompt will produce three recognisably different "voices".
- **(b)** For Q2 — did each model claim to know recent news, or did one or more say *"I don't have access to current information"*? This is the **knowledge cutoff** problem from the theory: a static LLM only knows what it was trained on. Note which models have a web-search tool turned on by default (recent answers with citation links) versus which are answering from the static weights only.
- **(c)** For Q3 — write down each model's stated cutoff date. Compare with the actual date today. The gap is the window the model literally cannot see — and it is why, for time-sensitive questions, you either need a model with tools (web, search) or a RAG system on top.

✓ You have now run a controlled comparison of the three leading consumer chat assistants on the same prompts — the most common "which one should I use" question, answered with evidence instead of vibes.

Task 2 — See RAG in action (NotebookLM)

Open notebooklm.google.com (a free Google account is enough). Click **New notebook**. Upload 2–3 PDFs you have lying around — a product manual, a research paper, a long article, a textbook chapter, a company handbook. If you have nothing handy, grab any free PDF from arxiv.org or a public-domain book from gutenberg.org.

Wait ~30 seconds for NotebookLM to ingest the documents. Then ask the chat panel two questions:

1. *"What are the three most important points in these documents?"*
2. A second, more specific question that **requires reading the documents** — e.g. *"What does section 3 of document 2 say about X?", "Which document mentions Y, and what does it claim?", or "Summarise the methodology paragraph on page 5 of the paper."*

Look closely at each answer. You'll see small numbered **citations** next to most claims, each linking back to the specific source passage. Click one and watch it jump you to the exact paragraph the model is drawing from.

Self-check — answer all three:

- **(a)** Where did the answer come from? Click two or three of the citations and note that each one points to a specific page in a specific document. *This is RAG made visible* — the model didn't pull the answer from its training weights; it pulled it from the chunks of your PDFs that the retriever fetched in response to your question.
- **(b)** Now ask a question whose answer is **not** in your uploaded documents (e.g. *"What will the FTSE close at next Friday?"* or *"What is the capital of a country none of these documents mention?"*). Notice what happens — does the model decline, hedge ("the sources don't say"), or hallucinate confidently? Compare this to how a plain chatbot would have handled the same question.
- **(c)** Write one sentence on why citations matter when an LLM is reading your private documents. (Hint: the answer is not "they look nice". It's about auditability — the ability for you, or a colleague, or a regulator, to check exactly which sentence in which document the model based its claim on.)

What you just saw

This is the same pattern almost every "chat with your docs" enterprise product is built on: chunk the documents, embed the chunks, retrieve the

most relevant ones at query time, and ask the LLM to answer *using those chunks as context*. The product changes; the recipe doesn't.

Task 3 — Watch an agent do a task

Open claude.ai (free or Pro account). The point of this task is to feel the difference between *a chatbot replies to you* and *an agent works on a task on your behalf*. Pick one of the two variants below:

Variant A — Lisbon trip planner (no engineering setup needed). Ask Claude:

"Plan a 3-day trip to Lisbon for a couple in October 2026, with a budget of €1500 (excluding flights). Include morning / afternoon / evening for each day, mix of culture and food, with one specific restaurant recommendation per evening. Return the result as a markdown table."

Watch how it works: it doesn't just dump an answer — on a multi-part request like this you'll typically see it think through the structure, list the days, then fill in the cells.

Variant B — coding agent (if you already have Cursor / Claude Code / Copilot installed). Open any small public GitHub repo locally and ask the assistant: "Add a [README.md](#) summarising what this repo does in 5 bullets." Watch the assistant read multiple files, propose a draft, and offer to write the file — that's an agent loop in miniature: read, reason, act.

Self-check — answer all three:

- **(a)** Did the model make any factual claims you'd want to verify before acting on them — a specific restaurant name, an opening time, a price, a quoted fact from the repo's source code? This is the **confabulation** point from the theory: even capable agents still need a human reviewing the output before it touches reality.
- **(b)** How many distinct steps did the model take to complete the request? In agentic tools you can often see the step list (Variant B will literally show file

reads and edits; Variant A will show a reasoning outline). Compare that to a single-shot chatbot reply — an agent decomposes the problem and works through it.

- **(c)** Write one sentence on the difference between *"a chatbot replies"* and *"an agent works on a task"*. The shorthand: a chatbot answers your message; an agent runs a small plan against an environment (files, web, APIs, your screen) and reports back.

Don't actually fly to Lisbon

For the purposes of this exercise you do **not** need to book a trip or merge any code. The point is the *experience* of watching a model decompose and execute a multi-step task, not the artefact it produces. Resist the urge to spend the next hour iterating the itinerary — close the tab when you can answer (a)–(c).

What "done" looks like

You're done when, in your notebook, you have:

- three screenshots of Q1 answers and short notes on style / cutoff / recent-news behaviour from Task 1 (chat assistants)
- two citation observations and one sentence on why citations matter from Task 2 (RAG)
- one verification-worthy claim, a rough step count, and one sentence distinguishing chatbot from agent from Task 3 (agents)

The aim was never to make you pick a favourite chatbot or to ship an agent in 45 minutes. It was to make sure that next time you read the words *"LLM"*, *"RAG"*, or *"agent"* in a product announcement or a job description, you have a concrete picture in your head of each — because you've now sat in front of all three and watched them behave.

Where this goes next

Module 4 ends here. The skills you just exercised — comparing model outputs, noticing what's grounded versus hallucinated, watching an agent step through a task — are the daily skills of working with LLMs in 2026. The technology will keep moving fast; the habit of *poking at* a system instead of trusting it is what stays useful.