ChatGPT Part 7: Building Custom Assistants for Your Team

A custom assistant, called a custom GPT in ChatGPT, is set up for one recurring job with its own instructions and reference files. Used well, it gives a team the same reliable starting point every time. Five steps build one that colleagues can trust: job, rules, sources, test and rollout. In this part, Karim, a purchasing manager at a packaging manufacturer, builds one for his buyers.

1. Job

An assistant that tries to do everything does nothing well, so we start with one job. We name one recurring task, the people who will use it and what a good answer looks like. The question is: which one task, for which users? We also write down what is out of scope, so nobody expects it to approve or decide. A narrow first version is easier to test and to trust.

Example: A finance team answers the same expense questions every week. It uses the Job question: which one task, for which users? It chooses one job, answering staff questions about expense policy, and a good answer quotes the policy and names the next step.

At work: Karim's buyers ask him the same questions all day, such as when a quote is needed. He uses the Job question: which one task, for which users? He chooses one job: answer buyers' questions about the purchasing policy, quote the section, and send everything else to him. Approving a purchase is out of scope.

2. Rules

Instructions are the standing orders that the assistant follows in every chat. We write them as short, plain lines: its role, the shape of an answer, what it must never do, and what it does when it is unsure. The question is: what must it always do, and never do? The most important rule is the fallback, which says to admit it and name a person when the sources have no answer. Behaviour belongs in the instructions, while facts belong in the files.

Example: A bank wants an assistant that explains card fees to its service desk. It uses the Rules question: what must it always do, and never do? The assistant must always quote the fee schedule, and it must never promise a refund.

At work: Karim first types: write instructions for a purchasing assistant. The result is generic, with no limits, so the assistant could guess a rule. Then he asks again with the Rules question: what must it always do, and never do? He asks for a job, a quoting rule, a never rule and a fallback, and he receives instructions he can paste straight in.

3. Sources

An assistant is only as reliable as the files behind it. We upload current, approved documents that the users are allowed to see, and we remove old drafts, duplicates and blurry scans. The question is: which approved files, and are they current? Because the files can show up in answers, confidential material stays out unless the workspace rules allow it. Every file also needs an owner who can replace it.

Example: A support team uploads its returns policy, but the folder also holds last year's draft. It uses the Sources question: which approved files, and are they current? It deletes the old draft, so the assistant quotes only the current rule.

This document is a property of Asaseeyat Academy - © Copy Right Protected.

At work: Karim gathers the purchasing policy, the approved supplier list and the table of approval limits. He uses the Sources question: which approved files, and are they current? He finds two older copies of the supplier list and a blurry scan, so he deletes them and adds the approval date to each file name. Now every answer can point to a current document.

4. Test

Before colleagues rely on the assistant, we test it in the preview panel. We try a typical question, a tricky question that mixes two topics, a question the files cannot answer, and a request that crosses a boundary. The question is: what happens with a tricky or unanswerable question? We compare each answer with the source and note every failure. After each fix, we run the same tests again, and then a real user tries it, because their wording will differ from ours.

Example: A bank's compliance team tests its assistant with a standard question, a mixed question and one that the files cannot answer. It uses the Test question: what happens with a tricky or unanswerable question? The assistant guesses the third answer, so the team adds a fallback rule and tests again.

At work: Karim tests four questions in the preview panel: when a quote is needed, whether an order can be split, an unlisted supplier, and a request to approve a rush purchase. He uses the Test question: what happens with a tricky or unanswerable question? The assistant approves the rush purchase, so he adds a rule that only the head of procurement can approve, and he runs all four tests again.

5. Rollout

Sharing is a decision, not a click. We choose the smallest audience that needs the assistant and the lowest permission that works: chat access lets people use it, while edit access lets them change it. What we can create and share depends on our plan and workspace settings, so we confirm it with an administrator. The question is: who needs access, and who owns it? The owner reviews the files, the instructions and the feedback after every policy change.

Example: An IT service desk plans to give a new assistant to the whole company on day one. It uses the Rollout question: who needs access, and who owns it? It starts with the support team, gives chat access only and names one owner.

At work: Karim shares the assistant with his six buyers only, with chat access, and keeps edit access for himself and one backup. He uses the Rollout question: who needs access, and who owns it? He names himself as owner and books a review after every policy update. When the approval limits change, he replaces the file and runs his four tests the same day.

Conclusion

In summary, a trustworthy team assistant is built in five steps. We give it one job, clear rules, approved sources, honest tests and a careful rollout with an owner. Before every release, the key question is simple: would we trust a colleague to act on every answer?

Sources

- OpenAI Help Center: Creating and editing GPTs – https://help.openai.com/en/articles/8554397-creating-and-editing-gpts
- OpenAI Help Center: Sharing and publishing GPTs – https://help.openai.com/en/articles/8798878-sharing-and-publishing-gpts
- OpenAI Help Center: GPTs in ChatGPT – https://help.openai.com/en/articles/8554407-gpts-in-chatgpt

- ChatGPT and OpenAI are trademarks of OpenAI. This video is independent training material and is not affiliated with OpenAI.

This document is a property of Asaseeyat Academy - © Copy Right Protected.