Take-Home Exercise: Build Your Own AI Copilot

ITAG Skillnet AI Advantage

Before You Begin

What you'll need: - A computer with internet access - Access to Langflow Cloud (https://astra.datastax.com/langflow) - free account - Access to ChatGPT or Claude for testing prompts - About 60-75 minutes of focused time - 2-3 text documents from your work (policies, FAQs, guides)

How to complete this exercise: 1. Work through each part in order - they build on each other 2. Follow the step-by-step instructions to build your copilot 3. Document your configuration choices as you go 4. Test your copilot with real questions from your work

What you'll learn: - How to create an AI copilot using no-code visual tools - How RAG (Retrieval-Augmented Generation) works in practice - How to configure chunking, embeddings, and retrieval - How to optimize your copilot for accuracy

---

PART 1: Prepare Your Knowledge Base

Time: 15 minutes

Before building your AI copilot, you need to prepare the documents it will learn from.

Exercise 1.1: Document Selection

What you'll do: Choose and prepare documents for your AI copilot.

STEP 1: Identify 2-3 documents from your work that would make a helpful assistant. Consider:

Document Type	Example Use Case
HR Policies	Answer employee questions about leave, expenses, benefits
Product FAQ	Help customers understand features and troubleshoot
Process Guide	Walk colleagues through standard procedures
Technical Docs	Answer questions about APIs, integrations, systems
Company Handbook	Onboarding info, values, organizational details

STEP 2: List your chosen documents:

Document	Topic	Size (approx.)	Why This Document?
1.	_____	_____	_____
2.	_____	_____	_____
3.	_____	_____	_____

Exercise 1.2: Document Preparation

What you'll do: Prepare your documents for optimal AI processing.

STEP 1: Review each document and answer:

Preparation Check	Doc 1	Doc 2	Doc 3
Clear section headings?	Yes/No	Yes/No	Yes/No
Outdated info removed?	Yes/No	Yes/No	Yes/No
In plain text or PDF format?	Yes/No	Yes/No	Yes/No
Under 50 pages?	Yes/No	Yes/No	Yes/No

STEP 2: If needed, clean up your documents: - Add clear headings if missing - Remove outdated sections - Convert to .txt or .pdf if in other formats - Break very long documents into logical sections

KEY INSIGHT: Better-structured documents = better AI answers. Time spent here pays off.

Exercise 1.3: Define Your Use Case

What you'll do: Clearly define what your copilot will help with.

STEP 1: Complete this use case definition:

Copilot Name: _____

Primary Users: (Who will ask questions?) - [ ] Employees - [ ] Customers - [ ] Partners - [ ] Other: ___

Types of Questions It Should Answer: 1. _____ **2.** _____ ***3.*** _____

Types of Questions It Should Decline: 1. _____ **2.** _____

Example Good Question: _____

Example Out-of-Scope Question: _____

PART 2: Build Your RAG Pipeline

Time: 25 minutes

Now you'll build your AI copilot using Langflow's visual interface.

Exercise 2.1: Set Up Langflow

What you'll do: Create a new RAG project in Langflow.

STEP 1: Go to https://astra.datastax.com/langflow and sign up/log in (free).

STEP 2: Click "New Flow" and select the **"Vector Store RAG"** template.

STEP 3: Take a screenshot or note what components you see:

Component	Purpose
File	_____
Text Splitter	_____
Embeddings	_____
Vector Store	_____
Retriever	_____
LLM	_____
Chat	_____

KEY INSIGHT: This is exactly the pipeline from our demo - documents flow through splitting, embedding, storage, and retrieval before reaching the LLM.

Exercise 2.2: Configure the Text Splitter

What you'll do: Set how your documents get broken into chunks.

STEP 1: Click the Text Splitter component and configure:

Setting	Recommended Value	Your Choice	Why This Setting?
Chunk Size	500 characters	_____	Smaller = specific answers
Chunk Overlap	50 characters	_____	Captures info across boundaries
Separator	\n\n (paragraphs)	_____	Respects document structure

STEP 2: Consider your content type:

- **For factual Q&A (policies, specs):** Use smaller chunks (300-500)
- **For contextual answers (guides, explanations):** Use larger chunks (800-1000)
- **For mixed content:** Start with 500 and adjust based on results

Your decision:

I'm using chunk size _ **because** _____

Exercise 2.3: Configure the Vector Store

What you'll do: Set up where your document embeddings will be stored.

STEP 1: Choose your vector store:

Option	Pros	Cons	Best For
Chroma (in-memory)	Free, simple, fast setup	Data lost on restart	Testing, demos
Chroma (persistent)	Free, data persists	Requires local storage	Development
Astra DB	Cloud-hosted, scalable	Requires account setup	Production

STEP 2: Configure your choice:

Vector Store Type: _____

Collection Name: _____ (Use something descriptive like "hr_assistant_docs" or "product_faq")

Exercise 2.4: Configure the Retriever

What you'll do: Set how many relevant chunks to retrieve for each question.

STEP 1: Configure retriever settings:

Setting	Value	Your Reasoning
Search Type	Similarity	_____
Top K (number of results)	3-5	_____

STEP 2: Understand the trade-off:

- **Lower K (2-3):** Faster, more focused answers, might miss relevant info
- **Higher K (5-7):** More context, better coverage, but potentially slower and noisier

Your decision:

I'm using Top K = _ **because** _____

Exercise 2.5: Configure the LLM

What you'll do: Choose and configure the language model that generates answers.

STEP 1: Select your LLM provider:

Provider	Model	Cost	Speed	Notes
OpenAI	gpt-4o-mini	$$	Fast	Great quality, requires API key
Groq	llama-3-70b	Free tier	Very fast	Good for testing
Ollama	Various	Free	Varies	Runs locally

Your choice: _____

STEP 2: Configure LLM settings:

Setting	Recommended	Your Choice	Why?
Temperature	0.1	_____	Lower = factual, higher = creative
Max Tokens	500	_____	Limits response length

STEP 3: Write your system prompt:

Use this template and customize for your use case:

```
You are a helpful assistant for [ORGANIZATION/DEPARTMENT].
Answer questions based ONLY on the provided context.
If the answer isn't in the context, say "I don't have information about that in my knowledge
base."
Always be professional and [SPECIFIC TONE - helpful/formal/friendly].
[ANY SPECIFIC INSTRUCTIONS FOR YOUR USE CASE]
```

Your system prompt:

```

```

PART 3: Test and Optimize

Time: 15 minutes

Now test your copilot and improve its performance.

Exercise 3.1: Upload and Index Your Documents

What you'll do: Upload your prepared documents and process them.

STEP 1: Upload your documents to the File component.

STEP 2: Click the Play button to run the indexing pipeline.

STEP 3: Note any issues:

Issue	Resolution
_____	_____
_____	_____

Exercise 3.2: Test Your Copilot

What you'll do: Ask questions and evaluate the answers.

STEP 1: Test with 5 questions from your expected use cases:

Question	Expected Answer (Key Points)	Actual Answer	Score (1-5)
1.	_____	_____	_____
2.	_____	_____	_____
3.	_____	_____	_____
4.	_____	_____	_____
5.	_____	_____	_____

Scoring Guide: - 5 = Perfect, accurate, complete - 4 = Good, mostly accurate, minor gaps - 3 = Acceptable, some inaccuracies - 2 = Poor, significant problems - 1 = Unusable, wrong or no answer

Average Score: _____ / 5

Exercise 3.3: Diagnose and Improve

What you'll do: Identify issues and make improvements.

STEP 1: Analyze your test results:

If You Noticed...	Likely Cause	Solution to Try
Answers too vague	Chunks too large	Reduce chunk size to 300-400
Missing obvious info	Top K too low	Increase to 5-7
Irrelevant info in answers	Top K too high	Reduce to 2-3
Answers too long	Max tokens too high	Reduce to 300
Wrong tone	System prompt	Refine tone instructions
Hallucinations	Temperature too high	Reduce to 0.0-0.1

STEP 2: Make ONE change and retest:

Change made: _____

New score: _____ / 5

Improvement: _____

Exercise 3.4: Test Edge Cases

What you'll do: Test how your copilot handles tricky situations.

Test Case	Your Question	Expected Behavior	Actual Behavior
Out-of-scope question	_____	Should decline gracefully	_____
Multi-part question	_____	Should address all parts	_____
Ambiguous question	_____	Should clarify or provide options	_____
Question with wrong assumption	_____	Should correct gently	_____

PART 4: Document Your Copilot

Time: 10 minutes

Create documentation for your copilot so others can use and maintain it.

Exercise 4.1: Complete the Copilot Spec

What you'll do: Document your configuration choices.

AI Copilot Specification

Name: _____

Purpose: _____

Owner: _____

Date Created: _____

Configuration Summary

Component	Setting	Value
Text Splitter	Chunk Size	_____
Text Splitter	Chunk Overlap	_____
Text Splitter	Separator	_____
Vector Store	Type	_____
Vector Store	Collection Name	_____
Retriever	Search Type	_____
Retriever	Top K	_____
LLM	Provider/Model	_____
LLM	Temperature	_____
LLM	Max Tokens	_____

System Prompt

>

Knowledge Base Documents

Document	Version	Last Updated
_____	_____	_____
_____	_____	_____
_____	_____	_____

Known Limitations

1. _____
2. _____

Maintenance Notes

- How often to update documents: _____
- Who can update the knowledge base: _____

Exercise 4.2: Plan for Improvement

What you'll do: Identify next steps to improve your copilot.

Short-term Improvements (This Week): 1. _____ **2.** _____

Medium-term Improvements (This Month): 1. _____ **2.** _____

Additional Documents to Add: 1. _____ **2.** _____

Wrap-Up: Reflection & Next Steps

What I Learned

STEP 1: Answer these questions:

1. The most challenging part of building my copilot was:

---

1. The configuration that had the biggest impact on quality was:

---

1. I was surprised that:

---

1. For my next copilot, I would do differently:

7-Day Challenge

Commit to improving and using your copilot this week:

- [ ] **Day 1:** Have a colleague test your copilot and give feedback
- [ ] **Day 2:** Add one more document to the knowledge base
- [ ] **Day 3:** Refine your system prompt based on user feedback
- [ ] **Day 4:** Test 10 new questions and note any gaps
- [ ] **Day 5:** Experiment with different chunk sizes
- [ ] **Day 6:** Share your copilot with your team
- [ ] **Day 7:** Document lessons learned and plan improvements

---

Resources for Continued Learning

- **Langflow Documentation:** https://docs.langflow.org/
- **RAG Best Practices:** https://www.pinecone.io/learn/retrieval-augmented-generation/
- **OpenAI Embedding Guide:** https://platform.openai.com/docs/guides/embeddings
- **Chroma Documentation:** https://docs.trychroma.com/

Quick Reference Card

RAG Pipeline Flow

```
Documents --> Split into Chunks --> Convert to Vectors --> Store in Database
                                                                    |
                                                                    v
    Answer <-- Generate with LLM <-- Retrieve Similar Chunks <-- User Query
```

Configuration Cheat Sheet

Goal	Chunk Size	Top K	Temperature
Factual Q&A	300-500	3-4	0.0-0.1
Explanations	600-800	4-5	0.1-0.3
Creative content	500-700	5-7	0.5-0.7

Troubleshooting Guide

Problem	Solution
"I don't know" for known info	Increase Top K, check document uploaded
Answers include wrong info	Decrease Top K, lower temperature
Answers too generic	Decrease chunk size, improve system prompt
Slow responses	Decrease Top K, use faster model
Inconsistent quality	Lower temperature, add examples to prompt

---

Congratulations on building your AI copilot! Keep iterating and improving based on real user feedback.

ITAG Skillnet AI Advantage - Build Your Own AI Copilot