Chapter 3 Interactive Labs User Guide

This guide explains how to run, navigate, and learn from the Chapter 3 Interactive Labs application. The app augments Chapter 3 with five hands-on HTML5 simulations focused on retrieval, chunking, RAG grounding, tool use, and multi-agent orchestration for production systems.

1. What this application is

Chapter 3 Interactive Labs is a single-page, offline HTML5 learning environment. It provides five toy-faithful simulations designed to help learners practice agentic workflows and observe system-level tradeoffs that static diagrams cannot show.

- Runs locally in any modern browser (Chrome/Edge/Firefox).
- No installation, accounts, or network required.
- Five labs correspond directly to Chapter 3 sections.
- Uses simulated outputs to teach workflow intuition (not a full LLM).

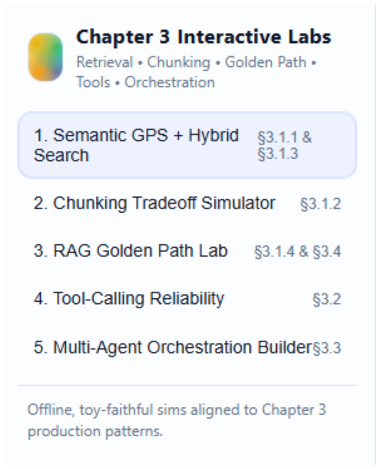
2. Getting started

2.1 Open the app

url

2.2 Navigation

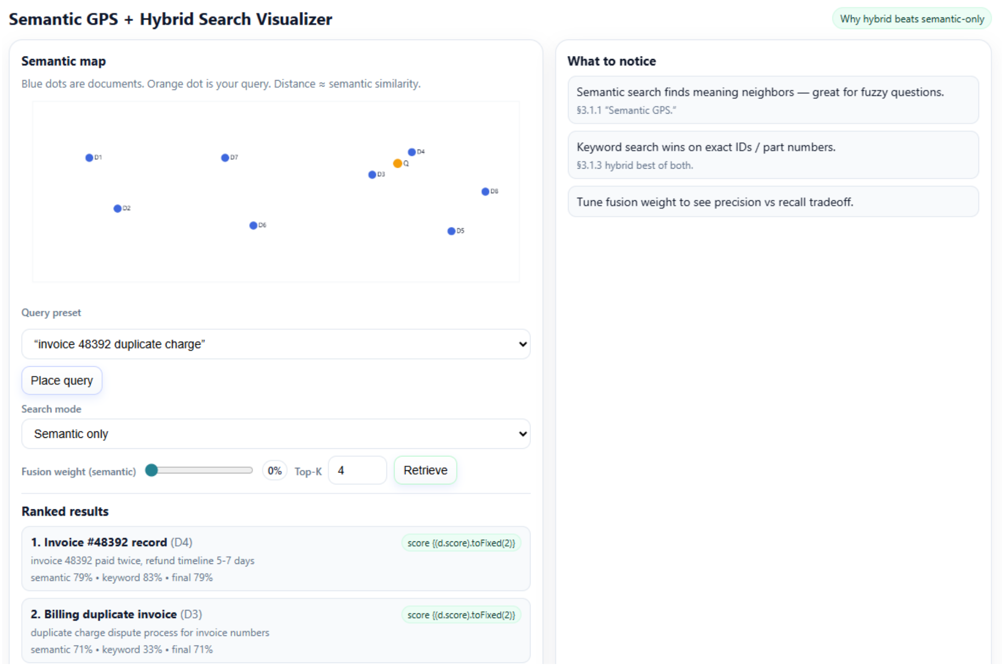
The left navigation menu lists the five labs. Click a lab title to switch views. On narrow screens the menu may hide; scroll to reach each lab section.


Chapter 3 Interactive Labs
Retrieval • Chunking • Golden Path • Tools • Orchestration
1. Semantic GPS + Hybrid Search §3.1.1 & §3.1.3
2. Chunking Tradeoff Simulator §3.1.2
3. RAG Golden Path Lab §3.1.4 & §3.4
4. Tool-Calling Reliability §3.2
5. Multi-Agent Orchestration Builder §3.3
Offline, toy-faithful sims aligned to Chapter 3 production patterns.

3. Lab-by-lab walkthrough

3.1 Lab 1 — Semantic GPS + Hybrid Search Visualizer (Chapter §3.1.1 & §3.1.3)

Purpose: Visualize embedding-space retrieval as a semantic 'map' and understand why hybrid search is required when exact keyword or ID matching matters.



Semantic GPS + Hybrid Search Visualizer

Why hybrid beats semantic-only

Semantic map

Blue dots are documents. Orange dot is your query. Distance ≈ semantic similarity.

Query preset: "invoice 48392 duplicate charge"

Place query

Search mode: Semantic only

Fusion weight (semantic) 0%   Top-K 4   Retrieve

Ranked results

1. Invoice #48392 record (D4)   score {(d.score).toFixed(2)}
invoice 48392 paid twice, refund timeline 5-7 days
semantic 79% • keyword 83% • final 79%

2. Billing duplicate invoice (D3)   score {(d.score).toFixed(2)}
duplicate charge dispute process for invoice numbers
semantic 71% • keyword 33% • final 71%

What to notice

Semantic search finds meaning neighbors — great for fuzzy questions.
§3.1.1 "Semantic GPS."

Keyword search wins on exact IDs / part numbers.
§3.1.3 hybrid best of both.

Tune fusion weight to see precision vs recall tradeoff.

How to use

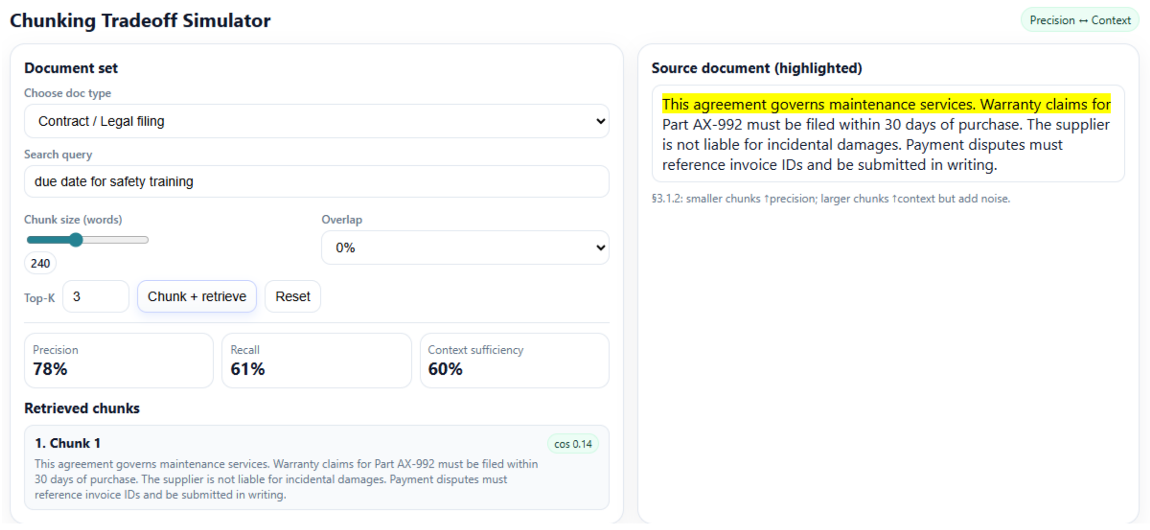
1. Select a Query preset (engine noise, invoice ID, part ID, safety policy).
2. Click "Place query" to set the orange query point on the semantic map.
3. Choose Search mode: Semantic only, Keyword only, or Hybrid.
4. If Hybrid, adjust Fusion weight to favor semantic vs keyword scoring.
5. Set Top-K.
6. Click "Retrieve."
7. Review the ranked results and the highlighted neighbors on the map.

How to interpret

- Semantic only excels on fuzzy 'meaning' questions but can miss exact identifiers.
- Keyword only is brittle for natural language but strong for IDs and part numbers.
- Hybrid lets you combine both; weight tuning shows precision/recall tradeoffs.
- Goal: internalize when to force hybrid in production.

3.2 Lab 2 — Chunking Tradeoff Simulator (Chapter §3.1.2)

Purpose: Experience how chunk size and overlap change retrieval precision, recall, and whether retrieved chunks include enough surrounding context.



Chunking Tradeoff Simulator

Precision ↔ Context

Document set

Choose doc type: Contract / Legal filing

Search query: due date for safety training

Chunk size (words): 240

Overlap: 0%

Top-K: 3

Chunk + retrieve | Reset

Precision 78%

Recall 61%

Context sufficiency 60%

Retrieved chunks

1. Chunk 1 — cos 0.14

This agreement governs maintenance services. Warranty claims for Part AX-992 must be filed within 30 days of purchase. The supplier is not liable for incidental damages. Payment disputes must reference invoice IDs and be submitted in writing.

Source document (highlighted)

This agreement governs maintenance services. Warranty claims for Part AX-992 must be filed within 30 days of purchase. The supplier is not liable for incidental damages. Payment disputes must reference invoice IDs and be submitted in writing.

§3.1.2: smaller chunks ↑precision; larger chunks ↑context but add noise.

How to use

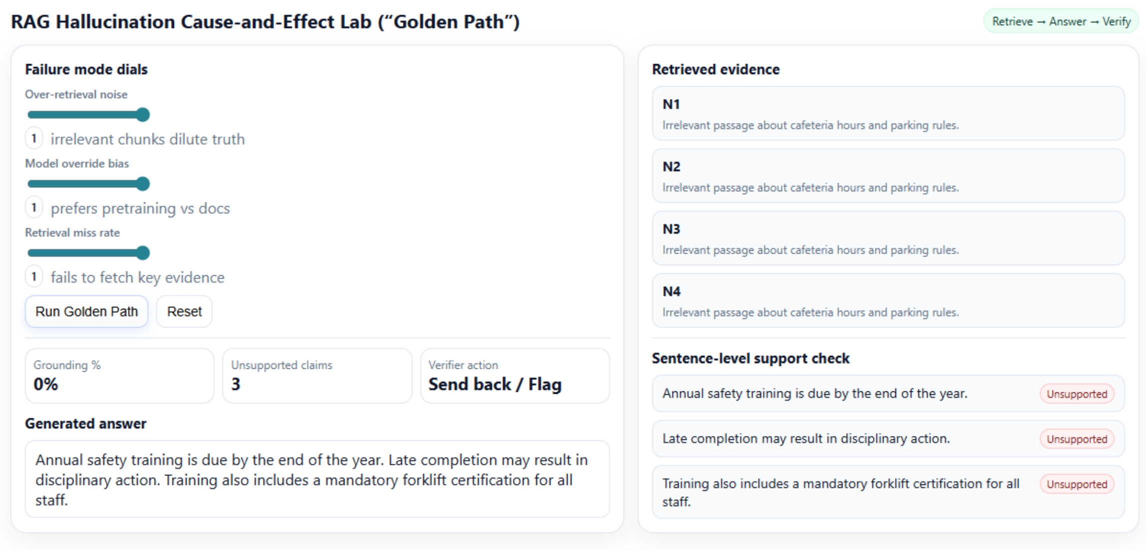
8. Choose a Doc type preset (policy, contract, tickets, manual).
9. Enter or edit the Search query.
10. Adjust Chunk size (words).
11. Choose Overlap (0%, 10%, 25%).
12. Set Top-K.
13. Click “Chunk + retrieve.”
14. Review retrieved chunks and highlighted spans in the source document.

How to interpret

- Precision rises with smaller chunks (less noise).
- Recall and context sufficiency typically rise with larger chunks.
- Overlap reduces boundary misses but increases token cost.
- Goal: learn that ‘best’ chunk size depends on document structure and query type.

3.3 Lab 3 — RAG Hallucination Cause-and-Effect Lab ("Golden Path") (Chapter §3.1.4 & §3.4)

Purpose: See why RAG can still hallucinate and how the Golden Path (Retrieve → Answer → Verify) mitigates risk.


RAG Hallucination Cause-and-Effect Lab ("Golden Path")

Retrieve → Answer → Verify

Failure mode dials

Over-retrieval noise: 1 — irrelevant chunks dilute truth

Model override bias: 1 — prefers pretraining vs docs

Retrieval miss rate: 1 — fails to fetch key evidence

Run Golden Path | Reset

Grounding %	Unsupported claims	Verifier action
0%	3	Send back / Flag

Generated answer

Annual safety training is due by the end of the year. Late completion may result in disciplinary action. Training also includes a mandatory forklift certification for all staff.

Retrieved evidence

N1 — Irrelevant passage about cafeteria hours and parking rules.

N2 — Irrelevant passage about cafeteria hours and parking rules.

N3 — Irrelevant passage about cafeteria hours and parking rules.

N4 — Irrelevant passage about cafeteria hours and parking rules.

Sentence-level support check

Annual safety training is due by the end of the year.	Unsupported
Late completion may result in disciplinary action.	Unsupported
Training also includes a mandatory forklift certification for all staff.	Unsupported

How to use

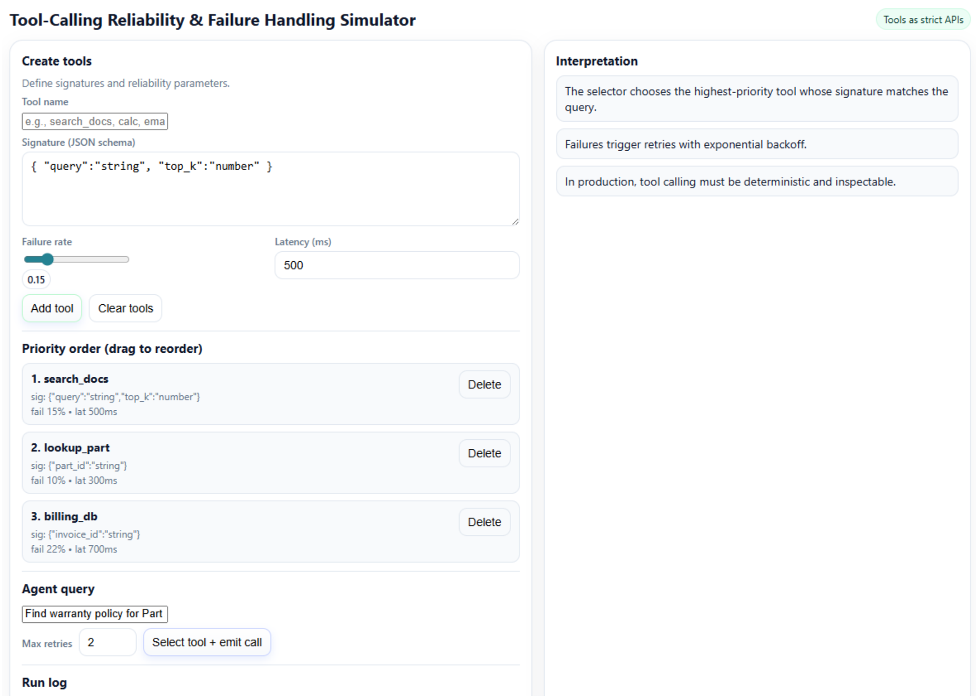
15. Set Over-retrieval noise to control irrelevant chunk load.
16. Set Model override bias to simulate the model preferring pretraining over evidence.
17. Set Retrieval miss rate to simulate missing key passages.
18. Click "Run Golden Path."
19. Compare Retrieved evidence, Generated answer, and Sentence-level support check.
20. Review KPI tiles (Grounding %, Unsupported claims, Verifier action).

How to interpret

- Higher noise dilutes evidence and increases unsupported claims.
- Higher bias makes answers drift from retrieved text even when evidence exists.
- Higher miss rate removes crucial support, forcing guesswork.
- Verifier action shows whether a last-stage checker would approve or send back.
- Goal: design RAG pipelines with explicit verification and low tolerance for unsupported claims.

3.4 Lab 4 — Tool-Calling Reliability & Failure Handling Simulator **(Chapter §3.2.1–§3.2.2)**

Purpose: Treat tools as deterministic APIs, not vague hints. Practice priority ordering, inspectable JSON calls, and retry/backoff behavior.


Tool-Calling Reliability & Failure Handling Simulator

Tools as strict APIs

Create tools

Define signatures and reliability parameters.

Tool name: e.g., search_docs, calc, ema

Signature (JSON schema)

```
{ "query":"string", "top_k":"number" }
```

Failure rate: 0.15

Latency (ms): 500

Add tool | Clear tools

Priority order (drag to reorder)

1. search_docs
   sig: {"query":"string","top_k":"number"}
   fail 15% • lat 500ms
   Delete
2. lookup_part
   sig: {"part_id":"string"}
   fail 10% • lat 300ms
   Delete
3. billing_db
   sig: {"invoice_id":"string"}
   fail 22% • lat 700ms
   Delete

Agent query

Find warranty policy for Part

Max retries: 2

Select tool + emit call

Run log

Interpretation

The selector chooses the highest-priority tool whose signature matches the query.

Failures trigger retries with exponential backoff.

In production, tool calling must be deterministic and inspectable.

How to use

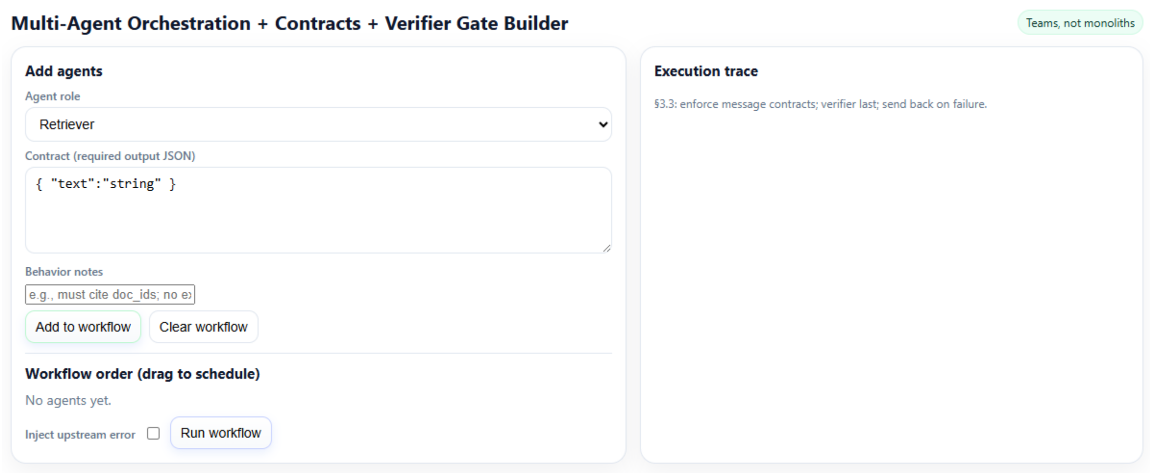
21. Enter a Tool name.
22. Paste a Signature as JSON (schema for arguments).
23. Set Failure rate and Latency.
24. Click “Add tool.” Repeat to add multiple tools.
25. Drag tools in the Priority list to reorder.
26. Enter an Agent query.
27. Set Max retries.
28. Click “Select tool + emit call.”
29. Review the Run log for emitted args, failures, and backoff.

How to interpret

- The selector chooses the highest-priority matching tool.
- Failures trigger retries with exponential backoff.
- Unmatched queries fall back to the first tool — a reminder to make routing explicit in production.
- Goal: build intuition for deterministic, observable tool use.

3.5 Lab 5 — Multi-Agent Orchestration + Contracts + Verifier Gate Builder (Chapter §3.3)

Purpose: Build a team of agents, enforce message contracts, and see why verifiers must run last.



Multi-Agent Orchestration + Contracts + Verifier Gate Builder

Teams, not monoliths

Add agents

Agent role: Retriever

Contract (required output JSON)

```
{ "text":"string" }
```

Behavior notes: e.g., must cite doc_ids; no e›

Add to workflow | Clear workflow

Workflow order (drag to schedule)

No agents yet.

Inject upstream error ☐ | Run workflow

Execution trace

§3.3: enforce message contracts; verifier last; send back on failure.

How to use

30. Select an Agent role (Retriever, Writer, Critic, Verifier, Guardrail).
31. Define a Contract as JSON specifying required output keys and types.
32. Optionally add Behavior notes.
33. Click “Add to workflow.” Repeat to build a chain.
34. Drag agents to reorder workflow scheduling.
35. Toggle “Inject upstream error” to simulate a wrong claim entering the chain.
36. Click “Run workflow.”
37. Review Execution trace for contract acceptance/rejection and verifier decision.

How to interpret

- Each handoff is a contract boundary; schema violations stop the chain.
- Injected errors show how later agents can miss faults unless verification is explicit.
- Verifier last enforces final trust before delivery.
- Goal: internalize multi-agent pipelines as orchestrated systems with strict interfaces.

4. Recommended learning activities

- Hybrid Search: Try the Part ID preset in semantic-only mode, then hybrid; note the rank change.
- Chunking: Run the same query with chunk sizes 80, 160, 320 and compare KPIs.
- Golden Path: Increase one failure dial at a time and record grounding % deltas.
- Tools: Add a new tool (e.g., ‘hr_policy_lookup’) and put it at different priorities; watch routing.
- Orchestration: Build Retriever→Writer→Verifier, then add Critic and Guardrail; compare outcomes.

5. Troubleshooting

- No UI appears: try Chrome/Edge or start a local server (Section 2.1).
- Sliders/buttons not responding: refresh the page (Ctrl+R).
- Drag-reorder not working: ensure you click-drag the card itself, not a text field.
- To host online: upload the folder to any static host (GitHub Pages, S3, Netlify).

6. Concept mapping back to Chapter 3

- §3.1 Retrieval Foundations → Lab 1
- §3.1.2 Chunking Strategy → Lab 2
- §3.1.4 Failure Modes + §3.4 Golden Path → Lab 3
- §3.2 Tools as APIs + Failure Handling → Lab 4
- §3.3 Multi-Agent Orchestration + Contracts + Verifier → Lab 5