3.3 RECAP & NEXT STEPS (2H DISCUSSION)

SESSION OVERVIEW

This is a 2-hour interactive recap and discussion to consolidate learning and plan next steps.

PART 1 – TIMELINE RECAP

Walk through the workshop story:

- **Day 0:** Environment setup
  - Two env repos (Ollama + watsonx).
  - `accelerator` repo.
  - `labs-src` reference notebooks.
- **Day 1:** LLM basics & prompting
  - Prompt patterns and reliability.
  - Compare local vs hosted models (Ollama vs watsonx).
- **Day 2:** RAG
  - Local RAG notebooks (Ollama + Chroma).
  - RAG notebooks in `labs-src` (Elasticsearch, Chroma, Granite).
  - Accelerator ingestion + retriever + pipeline skeleton.
- **Day 3:** Orchestration & agents
  - Agent notebook calling the accelerator API.
  - Accelerator API & (optional) Streamlit UI.
  - Governance & Orchestrate notebooks.

PART 2 – GROUP REFLECTION

Suggested questions for the group:

- What was most surprising across the three days?
- Biggest “aha” moments?
- Hardest parts of the labs?
- Where did you hit friction with:
  - Environments?
  - Libraries / SDKs?
  - Conceptual pieces (RAG, agents, governance)?

Capture answers on a shared board or document.

PART 3 – OPEN Q&A & TROUBLESHOOTING

Use this time to address remaining technical issues:

- Ollama / watsonx environments:
  - Model loading, API keys, network.
- Accelerator service:
  - `/ask` endpoint behaviour.
  - Vector DB connectivity.
  - Logging & configuration.
- Agent notebooks:
  - Tool selection.
  - JSON formatting issues.
  - Error handling.

Encourage participants to share **their code** and walk through solutions live.

PART 4 – FROM WORKSHOP TO REAL PROJECTS

Discuss how to turn the accelerator into something “real”:

- **Production microservice**
  - Hardened FastAPI app.
  - Config via `service/deps.py` and environment variables.
  - Health checks, observability, logging.
- **Scalable ingestion pipeline**
  - Batch extraction / chunking / embedding.
  - Scheduling (nightly or event-driven).
  - Monitoring index size and freshness.
- **Integration**
  - CI/CD pipelines using `Makefile`, `pyproject.toml`, Dockerfiles.
  - Deployment targets: Kubernetes / OpenShift / Cloud Foundry / VM.
  - Governance and evaluation workflows with watsonx.governance.

PART 5 – SUGGESTED NEXT STEPS

Potential directions for deeper dives:

- **Retrieval & ranking**
  - Hybrid search (lexical + semantic).
  - Rerankers and scoring strategies in `retriever.py`.
- **Prompting & safety**
  - Advanced prompt strategies in `prompt.py`.
  - Safety / refusal patterns and guardrails.
- **Multi-tenancy**
  - Tenant-aware configuration in `service/deps.py`.
  - Per-tenant indices and model settings.
- **UI & UX**
  - Richer Streamlit UI (`ui/app.py`).
  - Conversation history, feedback buttons, source highlighting.

Provide pointers to:

- Internal docs and repos.
- IBM public samples (`labs-src`, GitHub repos).
- Governance and Orchestrate documentation.

OPTIONAL: FEEDBACK FORM

Share a link or QR code to a short survey covering:

- Overall satisfaction.
- Clarity of explanations.
- Lab difficulty.
- Which topics participants want more of.

Use this to plan the **capstone day** and any follow-up sessions.