

Facilitator Notes: Agentic AI Workshop

90-Minute Workshop for 40 Participants

Pre-Event Checklist (1 Week Before)

Technical Setup

- **Test the complete agent end-to-end** (run through entire workshop yourself)
- **Create GitHub repository** with all code
- **Deploy a backup version** to Cloud Run (in case local fails)
- **Prepare 5 backup API keys** (for participants who have issues)
- **Test on both Windows and Mac** (if applicable)
- **Record a demo video** (backup if live demo fails)
- **Set up screen sharing** and test audio/video

Materials

- **Print orchestrator quick reference cards** (15 copies)
- **Create QR codes** for:
 - GitHub repository
 - AI Studio (<https://aistudio.google.com/>)
 - Feedback form
- **Prepare slides** with architecture diagrams
- **Create name tags** (optional: color-coded for orchestrators/implementers)

Communication

- **Send pre-event email** (see template below)
 - **Create Slack/Discord channel** for real-time help
 - **Prepare feedback form** (Google Forms)
 - **Test video conferencing** (if hybrid/virtual)
-

Pre-Event Email Template

Subject: Get Ready for Agentic AI Workshop - Action Required!

Body:

Hi everyone!

Excited to see you at the Agentic AI Workshop this [DATE] at [TIME]!

 **WHAT WE'LL BUILD:**

A complete enterprise AI agent with Google's stack (100% free!)

⚡ ACTION REQUIRED BEFORE THE EVENT:

1. GET YOUR FREE API KEY (5 min):
 - Go to: <https://aistudio.google.com/>
 - Click "Get API Key"
 - Save it somewhere safe
 - No credit card needed!

2. CLONE THE REPOSITORY:

```
git clone [YOUR-REPO-URL]  
cd insurance-agent-workshop
```

3. FOR CODERS (Implementers):

- Install Python 3.11+
- Run: cd backend && python -m venv venv
- Activate venv and run: pip install -r requirements.txt

4. FOR NON-CODERS (Orchestrators):

- Just have AI Studio open: <https://aistudio.google.com/>
- No installation needed!

📍 LOCATION: [Address]

🕒 TIME: [Time] (please arrive 10 min early)

💬 QUESTIONS: Reply to this email or join our Slack: [Link]

See you soon!

[Your Name]

⌚ Minute-by-Minute Timing Guide

0:00-0:15 | Setup & Introduction (15 min)

0:00-0:05 | Welcome & Logistics

- Welcome everyone, introduce yourself
- Point out restrooms, exits, WiFi password
- Explain workshop format: "Orchestrators and implementers work together"
- Set expectations: "Everyone leaves with a working agent"

Facilitator Script:

"Welcome! Today we're building an enterprise AI agent together. Half of you will be orchestrators - designing the agent's personality and behavior in AI Studio. The other half will be implementers - coding the logic. But here's the key: you'll work together throughout, not separately. Orchestrators design, implementers code it immediately, then we test together. By the end, we'll have a production-ready agent deployed to the cloud!"

0:05-0:10 | Quick API Key Check

- "Raise your hand if you already have your Gemini API key"
- For those who don't: "Open AI Studio now, click Get API Key"

- Have helpers assist anyone struggling

0:10-0:15 | Show the Final Product

- **CRITICAL:** Demo the complete working agent first
- Show conversation, knowledge search, quote calculation
- "This is what we're building in the next 75 minutes!"

Facilitator Tip: Build excitement by showing the end result first. People are more motivated when they see what they're working toward.

0:15-0:30 | Build Agent Personality (15 min)

0:15-0:20 | Orchestrators Design Prompt

- "Orchestrators, open AI Studio and create a new prompt"
- Show the prompt template on screen
- "Design Alex's personality - make them friendly and helpful"
- Walk around, help anyone stuck

0:20-0:25 | Implementers Code It

- "Implementers, create `system_prompt.py`"
- Show the code on screen
- "Copy the prompt from your orchestrator partner"
- Verify everyone has the file created

0:25-0:30 | Test Together

- "Implementers, run `python test_prompt.py`"
- "Orchestrators, does the response match your design?"
- If not: "Refine in AI Studio and update the code"

Facilitator Tip: Pair orchestrators with implementers sitting nearby. Encourage them to talk to each other!

Common Issues:

- **API key not working:** Use backup keys
 - **Import errors:** Check venv is activated
 - **Prompt too long:** Suggest simplifying
-

0:30-0:50 | Add Intelligence (20 min)

0:30-0:35 | Explain LangGraph

- Show architecture diagram
- "LangGraph lets the agent decide its next step"
- Draw on whiteboard: gather_info → calculate_quote
- "This is the 'Planning' principle in action"

0:35-0:40 | Orchestrators Create Test Cases

- "Write 5 test scenarios in AI Studio"
- Show test case format on screen
- "Think about: happy path, missing info, knowledge questions"

0:40-0:45 | Implementers Build Graph

- "Create `langgraph_agent.py`"
- "This is the most complex file - don't worry, it's pre-written"
- Walk through the code structure:
 - AgentState (memory)
 - Nodes (actions)
 - Conditional edges (decisions)

0:45-0:50 | Test Together

- "Run `python langgraph_agent.py`"
- "Orchestrators, verify your Test Case 1 passes"
- "If not, let's debug together"

Facilitator Tip: This is the hardest section. Be ready to help debug. Have the working code ready to copy/paste if needed.

Common Issues:

- **Graph doesn't compile:** Check indentation, imports
- **Tool not called:** Verify conditional logic
- **State not updating:** Check return statements

0:50-1:10 | Add Knowledge (20 min)**0:50-0:55 | Explain RAG**

- "RAG = Retrieval Augmented Generation"
- "Agent searches knowledge base before answering"
- Show diagram: Question → Vector Search → Context → LLM → Answer
- "This is the 'Learning' principle"

0:55-1:00 | Orchestrators Write FAQs

- "Write 10 insurance FAQs in AI Studio"
- Show FAQ template on screen
- "Cover: coverage types, discounts, general questions"

1:00-1:05 | Implementers Build RAG

- "Create `rag_system.py`"
- "Add orchestrator FAQs to INSURANCE KNOWLEDGE dict"
- "Run `python rag_system.py` to initialize"

1:05-1:10 | Test Together

- "Run `python test_rag.py`"
- "Ask: 'What is collision coverage?'"
- "Orchestrators, is the answer correct?"

Facilitator Tip: RAG can be slow to initialize. Warn participants it might take 30 seconds.

Common Issues:

- **Embeddings fail:** Check API key, internet connection
 - **No results found:** Verify knowledge base initialized
 - **Wrong answers:** Check FAQ content, refine queries
-

1:10-1:25 | Polish & Deploy (15 min)

1:10-1:15 | Create FastAPI Backend

- "Create `main.py`"
- "This ties everything together"
- "Run `python main.py`"
- "Open `http://localhost:8000/docs`"

1:15-1:20 | Final Testing

- "Orchestrators, test all 5 scenarios"
- Use curl commands or Postman
- "Document what works and what doesn't"

1:20-1:25 | Deploy to Cloud Run

- "Implementers, run the deploy command"
- Show command on screen
- "This takes 2-3 minutes"
- While deploying: "Let's review what we built"

Facilitator Tip: Start deployment early. If it fails, use the backup deployed version.

Common Issues:

- **Port already in use:** Kill other processes
 - **Deploy fails:** Use backup URL
 - **Slow deployment:** Show backup while waiting
-

1:25-1:30 | Wrap-Up (5 min)

1:25-1:27 | Celebrate Success

- "Everyone, test the live agent at [URL]"
- "You built a production AI agent in 90 minutes!"

- Take a group photo (optional)

1:27-1:29 | Share Resources

- Show QR codes for:
 - GitHub repo
 - Documentation
 - Community
 - Feedback form
- "Star the repo if you found it helpful!"

1:29-1:30 | Q&A

- "Any questions?"
- "What surprised you most?"
- "What will you build next?"

Facilitator Script:

"Congratulations! You just built an enterprise-grade AI agent using Google's stack. Orchestrators designed the personality, test cases, and knowledge base. Implementers coded the LangGraph workflow, RAG system, and deployed to production. Together, you created something neither group could have built alone. That's the power of collaboration!"

⚠ Troubleshooting Guide

Issue: "My API key doesn't work"

Solution:

1. Check it starts with `AIza`
2. Verify no extra spaces
3. Try regenerating in AI Studio
4. Use backup key if needed

Issue: "Import errors / module not found"

Solution:

1. Check venv is activated (see `(venv)` in terminal)
2. Run `pip install -r requirements.txt` again
3. Verify Python version: `python --version` (need 3.11+)

Issue: "LangGraph won't compile"

Solution:

1. Check all nodes are defined
2. Verify conditional edges return valid node names
3. Copy working code from GitHub

Issue: "RAG returns no results"

Solution:

1. Delete `insurance_knowledge_db` folder
2. Run `python rag_system.py` to reinitialize
3. Check embeddings API is working

Issue: "Agent doesn't remember context"

Solution:

1. Verify session management in `main.py`
2. Check `AgentState` includes messages
3. Ensure messages are appended, not replaced

Issue: "Deployment fails"

Solution:

1. Use backup deployed URL
 2. Check `gcloud` is installed and authenticated
 3. Verify Dockerfile is correct
 4. Run locally instead
-

Backup Plans

If Code Doesn't Work

- **Plan A:** Use pre-deployed version, focus on concepts
- **Plan B:** Show recorded demo, explain architecture
- **Plan C:** Turn it into a design workshop (AI Studio only)

If Time Runs Short

- **Skip deployment**, focus on local testing
- **Skip RAG**, focus on LangGraph
- **Skip frontend**, use curl for testing

If Participants Struggle

- **Pair beginners with experienced folks**
- **Have helpers circulate**
- **Slow down, skip optional sections**

If Internet Fails

- **Use hotspot** for critical demos
- **Have offline docs** ready
- **Focus on concepts** over live coding

Managing 40 Participants

Room Setup

- **Arrange tables** for pairs (orchestrator + implementer)
- **Have power strips** at each table
- **Set up projector** visible from all seats
- **Test audio** before starting

Helpers

- **Recruit 2-3 helpers** to circulate
- **Assign zones:** front, middle, back
- **Give them walkie-talkies** or Slack channel
- **Brief them** on common issues

Engagement

- **Ask questions** throughout: "Raise hand if your test passed"
- **Celebrate milestones:** Applaud when agent works
- **Share success stories:** "Table 5 just got their agent working!"
- **Keep energy high:** Use humor, take breaks if needed

Timing

- **Use a timer** visible to you
 - **Have a co-facilitator** manage time
 - **Be flexible:** Skip sections if behind
 - **Prioritize:** Core features > nice-to-haves
-

Pro Tips

Before the Workshop

1. **Arrive 30 min early** to test everything
2. **Have backup laptop** ready
3. **Print extra reference cards**
4. **Test the WiFi** with 40 devices
5. **Charge your devices**

During the Workshop

1. **Repeat questions** so everyone hears
2. **Use the microphone** even in small rooms
3. **Walk around** while people code
4. **Acknowledge struggles:** "This part is tricky"
5. **Celebrate small wins:** "Great question!"

After the Workshop

1. **Send follow-up email** with resources
 2. **Share photos** on social media
 3. **Collect feedback** via form
 4. **Plan next workshop** based on feedback
 5. **Stay available** for questions
-

Success Metrics

During Workshop

- **80%+ complete setup** (API key, clone repo)
- **60%+ get agent working** locally
- **40%+ deploy** to Cloud Run
- **Everyone understands** the 6 principles

Post-Workshop

- **90%+ would recommend** to a colleague
 - **70%+ feel confident** building simple agent
 - **50%+ plan to build** their own agent
 - **30%+ join** the community
-

Facilitator Scripts

Opening

"Welcome to the Agentic AI Workshop! I'm [Name], and I'm excited to help you build your first enterprise AI agent today. We have 90 minutes to go from zero to a deployed, production-ready agent. It sounds ambitious, but I promise you'll have a working agent by the end. Let's get started!"

Transitions

minutes for this part."

When Things Go Wrong

"Okay, I see some folks are hitting an error. That's totally normal - debugging is part of development. Let me show you how to fix it..."

Celebrating Success

"Wow, I'm seeing a lot of hands up - your agents are working! Give yourselves a round of applause. This is exactly what we wanted to see."

Closing

"You did it! In 90 minutes, you went from nothing to a deployed AI agent. More importantly, you learned how orchestrators and implementers work together. Take this knowledge and build something amazing. Thank you!"

Post-Event Checklist

- **Send thank you email** with resources
- **Share photos/videos** on social media
- **Collect feedback** via form
- **Review what worked/didn't**
- **Update materials** for next time
- **Follow up** with participants who had issues
- **Plan next workshop** (if applicable)

Additional Resources for Facilitators

Recommended Reading

- LangGraph documentation
- Google AI Studio guides
- Prompt engineering best practices

Community

- Join facilitator Slack channel
- Share lessons learned
- Get help from other facilitators

Continuous Improvement

- Record the workshop (with permission)
- Review recording for improvements
- Update materials based on feedback
- Test new features before next workshop

You've got this! The workshop will be amazing. 

Questions? Contact: [Your email/Slack]

END OF FACILITATOR NOTES