SESSION 11 REFERENCE GUIDE

Session 11: Iterate and Polish

AI + Research Level 2 — Supplementary Material

Concept reinforcement: THE EXPERIMENTATION LOOP
Space: No new Space — students improve their Session 10 projects
Pre-session prep: Open each student's Space URL in a browser tab. Have the Session 10 rescue templates ready for anyone who doesn't have a working Space.

Time Breakdown (2 hours)

0:00–0:15 — Quick Demos

Each student shows their Space to the group. ~2 minutes per student.

Format for each demo:
1. What does it do? (one sentence)
2. Show it working (one example input)
3. What's one thing you'd improve?

Instructor role: After each demo, lead the group in:
- One compliment ("I like that you...")
- One suggestion ("What if you tried...")

Keep it positive and fast. The goal is to give everyone a moment in the spotlight, not deep critique.

If a student doesn't have a working Space:
- No judgment. "That's okay — we'll get you set up in the next block."
- Pair them with the rescue templates from Session 10 (`session-10/templates/`)
- They'll fork a template and customize it during the 1-on-1 time

0:15–1:15 — 1-on-1 Feedback and Improvement

~10 minutes per student. Instructor shares screen and works on each student's Space together.

Rotation plan (5 students):

Slot	Time	Student	Others doing...
1	0:15–0:25	Student A	Testing each other's Spaces
2	0:25–0:35	Student B	Testing each other's Spaces
3	0:35–0:45	Student C	Testing each other's Spaces
4	0:45–0:55	Student D	Testing each other's Spaces
5	0:55–1:05	Student E	Testing each other's Spaces
Buffer	1:05–1:15	Catch-up	Finish any incomplete work

For each student, check this improvement list:

1. **Title and description** — Does the title make sense to someone who's never seen it? Does the description explain what it does?
2. **Examples** — Are there at least 2-3 example inputs? Do they showcase what the Space does well?
3. **Output formatting** — Is the output readable? Should it include labels, percentages, explanations?
4. **Input validation** — What happens with empty input? Very long input? Weird characters?
5. **Model choice** — Is the model appropriate for the task? Is it fast enough on free CPU?
6. **Model card link** — Add a link to the model card in the description so users can learn more.

Common fixes to make live:
- Improve the `placeholder` text in `gr.Textbox` to guide users
- Add more `examples` to the `gr.Interface`
- Add input length checking (`text[:512]`) to prevent timeouts
- Improve the function's return string to be more human-readable
- Fix the README.md to accurately describe the Space

For students without a working Space:
1. Pick a rescue template from Session 10
2. Fork it to their account
3. Change the title, description, model, and examples
4. Deploy and verify it works
5. They'll have a working Space for demo day even if it's simpler

1:15–1:40 — Peer Testing

Students swap and try each other's Spaces. Share the peer feedback form (see `peer-feedback-form.md`).

Instructions to students:
1. Open another student's Space URL
2. Try 3-4 inputs — at least one "normal" input and one "weird" input
3. Fill out the feedback form for each Space you test
4. Be specific — "the output was confusing" is less helpful than "I didn't know what POSITIVE (87%) meant"

Instructor role: Float between students, help with any technical issues, encourage specific feedback.

If time allows: Have each student read their feedback aloud to the Space owner. This creates accountability and surfaces the best suggestions.

1:25–1:40 — Notebook Time

Share the Colab notebook link in the Zoom chat. Students open it and try the debug challenges.

What they do:
- Run the setup cell
- Review the improvement checklist and common bugs table
- Try to find and fix the bug in Debug Challenge 1 (result is a list, not a dict — needs `result[0]`)
- Try to find and fix the bug in Debug Challenge 2 (empty string crashes the model — needs an `if not text` check)
- Fill in the "My Improvements" section with what they changed tonight

Instructor role: Let students struggle for a couple minutes before giving hints. The goal is recognizing error patterns, not speed.

Hints if stuck:
- Challenge 1: "What type is `result`? Try printing it before the return line."
- Challenge 2: "What happens when you pass an empty string to the model? How would you check for that?"

GitHub skill: Show how to edit a file directly on github.com (pencil icon). Have each student update their README or another file in their repo.

1:40–1:55 — Prep for Demo Day

Explain the format for next week:

Demo Day format (5-7 minutes per student):
1. **What it does** — One sentence. Who is it for?
2. **Live demo** — Show it working with 2-3 inputs
3. **What model** — Which model did you use and why?
4. **What went wrong** — What was the hardest part? What broke?
5. **What you'd improve** — If you had another month, what would you change?

Tips for presenting:
- Practice with a friend or family member before next session
- Have your inputs ready — don't search for examples during the demo
- It's okay to show something that doesn't work perfectly. Talking about what broke is more interesting than a flawless demo.
- You're not being graded. This is about showing what you built and what you learned.

Say: "You've all built something real that lives on the internet. Next week you get to show it off. Bring your best inputs and your best story about what went wrong."

1:55–2:00 — Between-Session: Final Polish

Share the between-session challenge. This is the one session where between-session work really matters.

Say: "This is the last between-session before demo day. If there's one thing you want to fix, now's the time. Practice your demo — 5 to 7 minutes, out loud, to yourself or someone else."

What Could Go Wrong

Problem	Fix
Student doesn't have a working Space	Use rescue templates from Session 10. Fork, customize, deploy.
10 minutes per student isn't enough	Prioritize the biggest impact fix (usually title + description + examples). Save code-level fixes for after session.
Peer feedback is too vague ("it's good")	Model specific feedback first: "I like that the examples show different use cases. I'd add a line explaining what the confidence percentage means."
Student is discouraged by feedback	Reframe: "Every suggestion is a chance to make it better. Professional developers get code reviews every day."
Spaces are asleep (cold start)	Open all student Space URLs 15 minutes before session to wake them up.
Student finished early during 1-on-1 rotation	Have them help test another student's Space or add more examples to their own.

Key Vocabulary (reinforce from previous sessions)

- **Iteration** — improving something through repeated cycles of testing and fixing
- **User testing** — having someone who didn't build it try to use it
- **Edge case** — an unusual input that the builder didn't think of
- **UX (user experience)** — how easy and clear the Space is to use