Isotopic Suite — Help Centre

Teacher walkthroughs in workflow order: create a Module, build resources in Lab, run them from Workroom, then Playground and Revision. Share this PDF with colleagues learning the Suite.

Generated 27 July 2026, 9:00 pm (AWST) · 17 guides

Contents

GUIDE 1 OF 17 · CLASS HUB — FOLLOW IN ORDER

Modules

Start here — create a Module on the class home, then build and assign activities inside it.

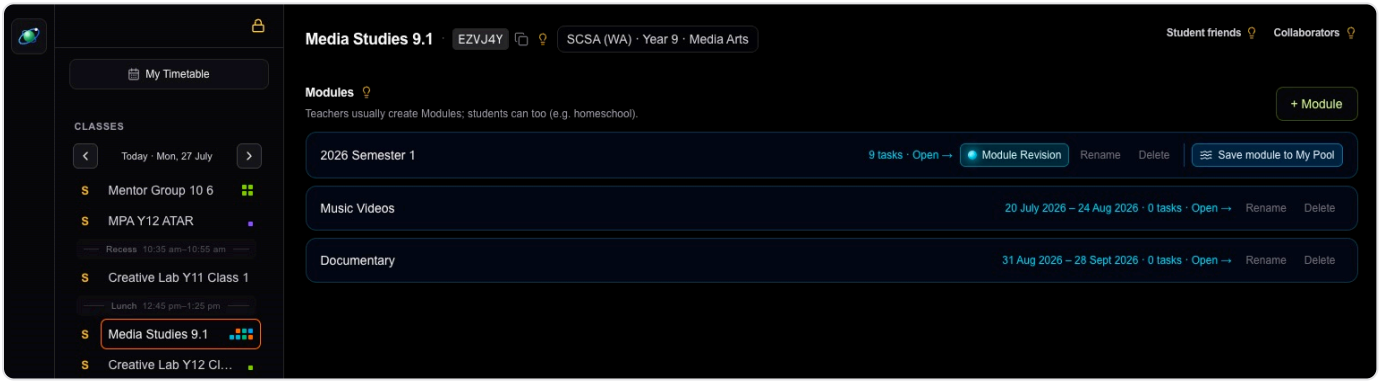
Where to start

Before you make a single activity, create at least one Module on the class home. Lab, Workroom, Playground, and Revision all live inside a Module — not on a flat class list.

A typical path looks like this: create a Module on the class home → open it → use Lab to build Echo, Spark, or Flux (and other apps) → assign so tasks appear in Workroom → students practise privately in Playground when needed → use Revision to recap the Module → optionally save the whole Module to My Pool for next year.

What Modules are for

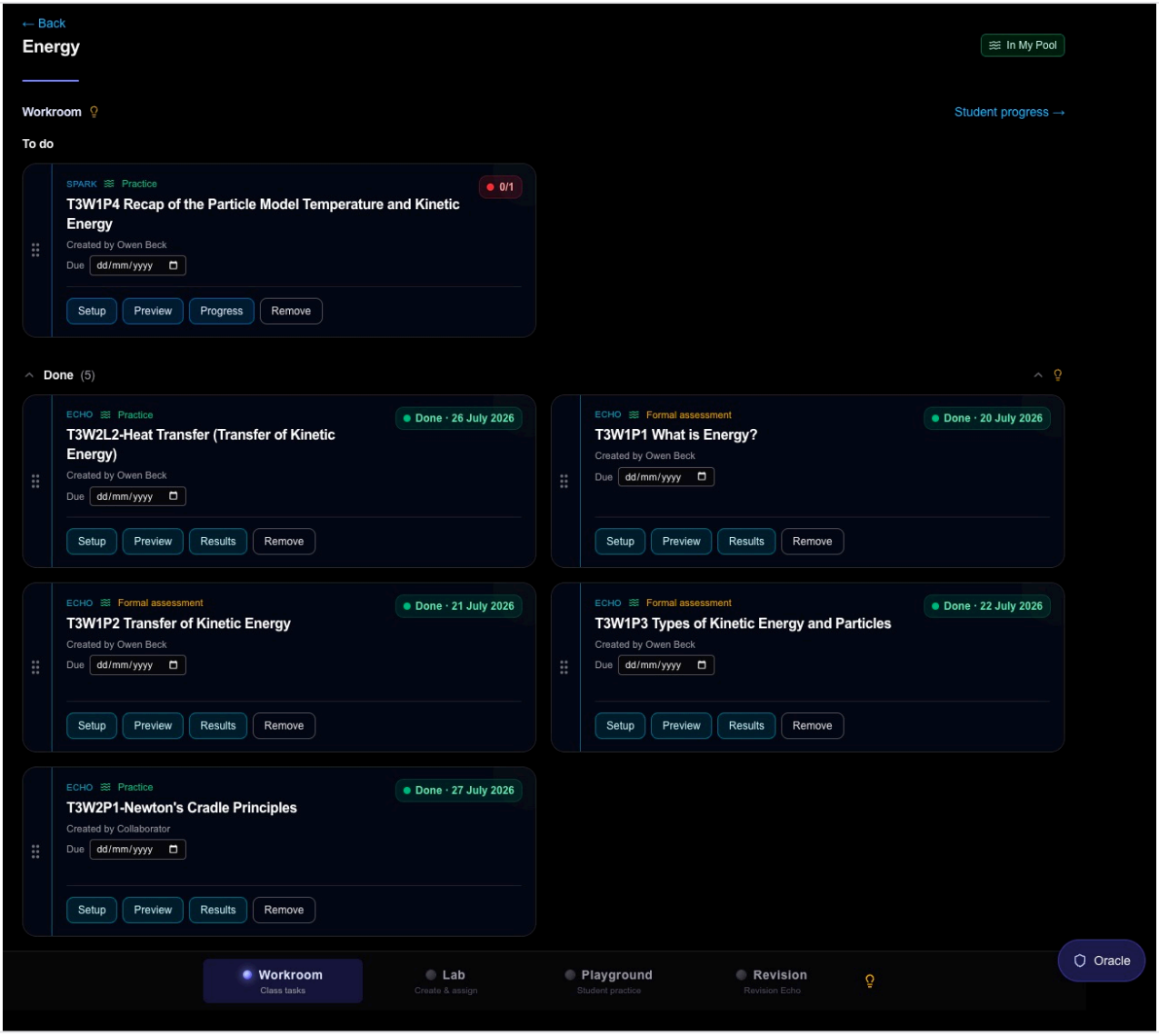
A Module is a shared topic shelf for your class — for example “Term 1 — Narrative”, “OLNA English”, or “Photosynthesis week”. Teachers and students see the same Module list on the class home.



Class home — shared Modules for this class

When you open a Module, a strip across the top takes you to four places. Workroom holds class-assigned tasks. Lab is where you create new activities for this topic. Playground is where students keep private practice scoped to this Module. Revision builds one Echo quiz from everything in the Module — class tasks plus Playground practice.

Surface	Who uses it	What it is for
Workroom	Everyone	Published class tasks for this Module
Lab	Teachers (Student Lab for students)	Create activities that land in Workroom or Playground
Playground	Students	Private practice packs tied to this Module
Revision	Everyone	A private Echo recap built from this Module’s tasks



Inside a Module — Workroom, Lab, Playground, and Revision

Creating and organising Modules

1. On the class home, choose Create Module and walk through the wizard.
2. Give the Module a clear name. You can optionally set start and finish teaching dates.
3. Open the Module — you land on Workroom. Use the strip to reach Lab, Playground, or Revision.
4. Your next step is Lab: create your first activity inside this Module (see the Lab guide).
5. To rename or delete, use the class home (the creator or staff can do this). Deleting a Module removes the shelf; tasks stay in the class but are no longer grouped there.

Teachers usually create Modules; students can too (for example in homeschool settings). Skipping this step is the most common setup mistake — activities need a Module to land in.

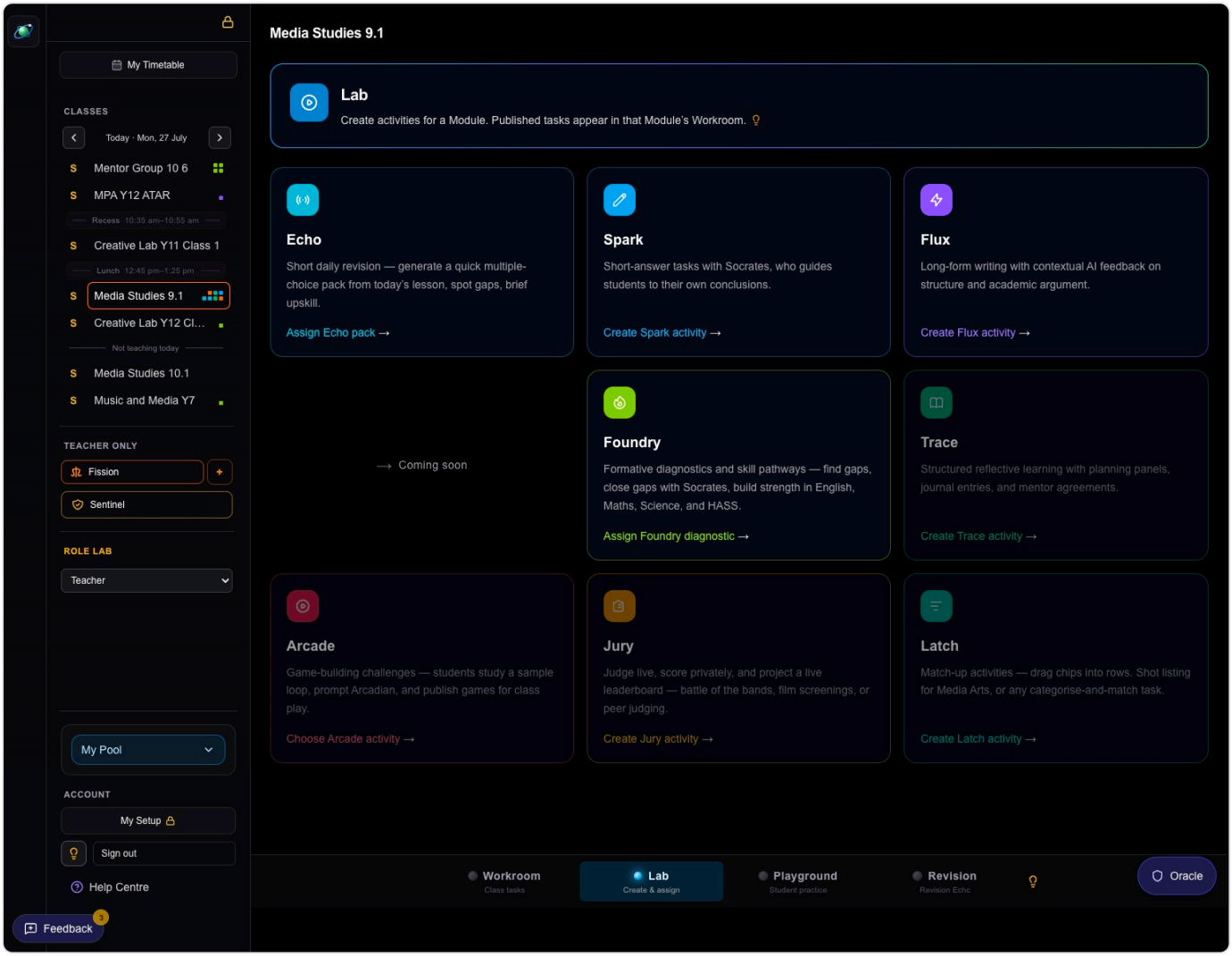
GUIDE 2 OF 17 · CLASS HUB — FOLLOW IN ORDER

Lab

Inside a Module, Lab is where you create Echo, Spark, Flux, and other activities — then assign them to that Module's Workroom.

What Lab is for

Lab is your create bench — but only inside a Module you have already set up. You design and publish activities here; the class actually sits them from that Module's Workroom.



Lab hub — choose a tool card to create

As a teacher, you pick an app card (Echo, Spark, Flux, and so on), walk through setup, and assign the finished task into the Module you are teaching. It then appears on that Module's Workroom for the class. Students do not create class work from Lab — they use Student Lab inside a Module when they want private practice, which lands in that Module's Playground.

- Lab is for creating and publishing learning activities
- Workroom is where assigned class tasks live day to day — not Lab
- Marking and moderation sit in Fission; the floating class AI helper is Oracle — neither is created from Lab

Where Lab fits in the Module

Think of the Module as the topic shelf and Lab as the workshop on that shelf. You create here; Workroom is the class noticeboard for what you published; Playground is where students keep private practice tied to the same topic.

Surface	What happens here
Module	The shared topic shelf — create this first on the class home
Lab	Choose an app and build an activity inside the Module
Workroom	Class-assigned tasks you published from Lab
Playground	Student private practice in the same Module

How you use Lab

1. Pick a class in the sidebar. You should already have at least one Module (see the Modules guide).
2. On the class home, open the Module you are teaching this week.
3. Use the Module strip → Lab, or open Lab from the sidebar while scoped to that Module.
4. Click a tool card — Echo, Spark, Flux, Foundry, and so on.
5. Work through setup. The Module is usually already selected; choose Practice or assessment, an optional due date, whether Oracle is allowed, and how Socrates should coach.
6. Publish or assign. The task card appears in this Module's Workroom.
7. Students open the same Module → Workroom to sit the work. You chase submissions and open results from the Workroom cards.

> You have it right when you created inside a Module, assigned the task, and the card shows in that Module's Workroom for you and your students.

What you can create from Lab

The main apps sit at the front of the Lab shelf. Others appear after a Coming soon marker — some are already usable there (such as Foundry). Each guide below walks through one app in more detail.

App	What it is for
Echo	Short daily/weekly revision MCQ packs from a lesson
Spark	Short-answer tasks with Socrates coaching
Flux	Long-form writing with target word count, structure, and argument feedback
Foundry	Formative diagnostics and skill pathways (Assay / Temper / Quench) — on the Coming soon shelf but usable
Trace	Planning and reflective journalling, with mentor agreements — Coming soon
Arcade	Human–AI–human creative/critical cycle — build games, judge with Jury, play in Playground — Coming soon
Jury	Live judging, private scores, projected leaderboard — Coming soon
Latch	Match-up / shot-listing — drag chips into rows — Coming soon

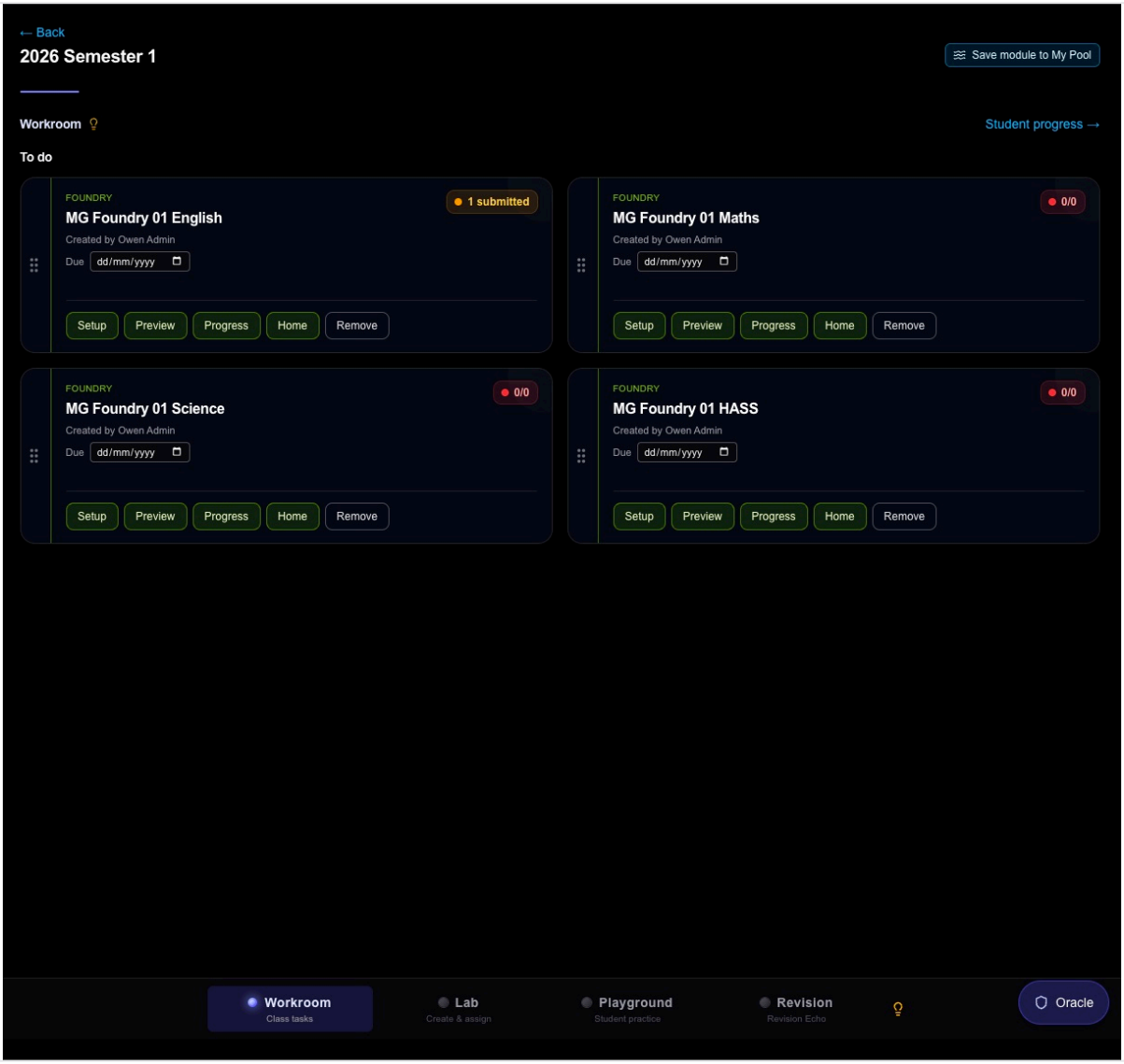
GUIDE 3 OF 17 · CLASS HUB — FOLLOW IN ORDER

Workroom

After you publish from Lab, assigned class tasks live here — inside the Module's Workroom.

What the Workroom is

The Workroom is inside each Module. It lists class-assigned work for that topic — not private Playground practice.



Workroom — published class tasks

When you create something in Lab and assign it into a Module, the card appears here. Students open the same cards to sit the work. You manage drafts, chase submissions, and open results from these cards. Module Revision (on the strip) can later build an Echo recap from Workroom tasks plus Playground practice.

What teachers do in the Workroom

Each task card gives you the actions you need for that stage of the work. To do cards focus on chasing who has not finished; Done cards focus on reviewing what was submitted.

- Open or continue setup if the task is still a draft
- Preview the session exactly as students will see it

- On To do cards, use the coloured completion chip (red, amber, or green) or open Progress to see who still needs to submit
- On Done cards, open Results to read finished work and diagnostics
- Open Student progress for a class-wide board — who has finished each task — then drill into Details for one task
- From Foundry cards, open Progress or Results for that diagnostic, or Foundry home for pathways, class gaps, and Socrates practice
- Remove sends a task to Recently removed for seven days; Restore brings it back with student work intact, then it is permanently deleted
- The class header shows who created a task and which collaborators are attached
- Due dates, when you set them at assign time, show on Workroom cards (Echo, Spark, Flux, Foundry)
- Fission-related cards can appear when moderation work is linked to the class

Think of the Workroom as your teaching hub for that Module — not only where students launch tasks, but where you review what they produced. Back from Progress, Results, or Student progress always returns you here.

Fission is not created from Lab, but related moderation cards can still show in the Workroom when they are relevant to the class.

Workroom, Lab, and Playground

Surface	Contains
Lab	Where you create and choose tools
Workroom	Class-assigned published tasks
Playground	Student private practice for the same Module

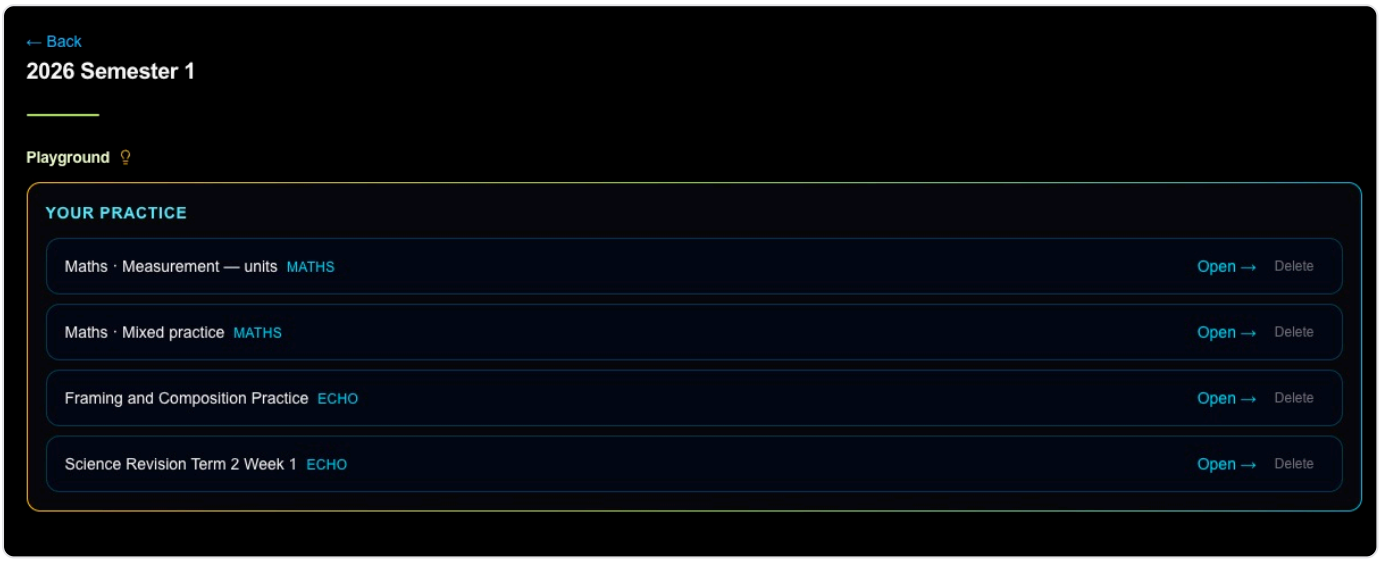
GUIDE 4 OF 17 · CLASS HUB — FOLLOW IN ORDER

Playground

Students' private practice shelf inside a Module — separate from class-assigned Workroom tasks.

What Playground is for

Playground is where students practise on their own, inside the same Module you are teaching. Class-assigned work stays in Workroom; Playground is for private packs they create from Student Lab.



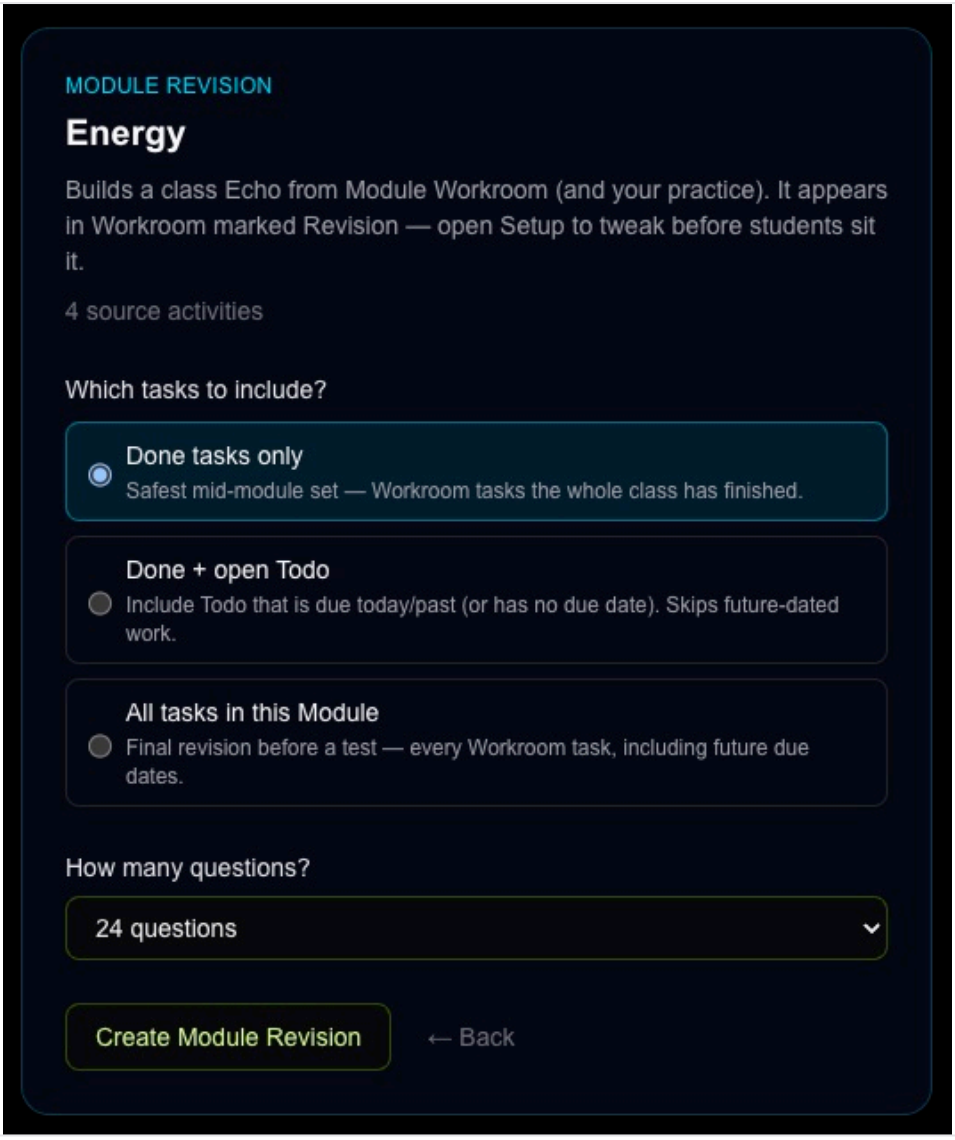
Playground — subject rooms, practice, and Modules

From Student Lab (scoped to a Module), a student can spin up their own Echo, Spark, or Flux practice. Those items land in Module Playground — not on the class Workroom. When the Module has enough work in both places, Revision can combine Workroom and Playground into one private Echo recap.

Module Revision

Once students have sat Workroom tasks and/or built Playground practice in a Module, open Revision on the Module strip. The Suite generates one private Echo quiz from everything on that topic shelf — class assignments plus personal practice. Each student's sit is private.

Use Revision at the end of a unit, before a test, or as a weekly recap. It complements — does not replace — Echo packs you assign from Lab.



Module Revision — private Echo from this Module's tasks

GUIDE 5 OF 17 · CLASS HUB — FOLLOW IN ORDER

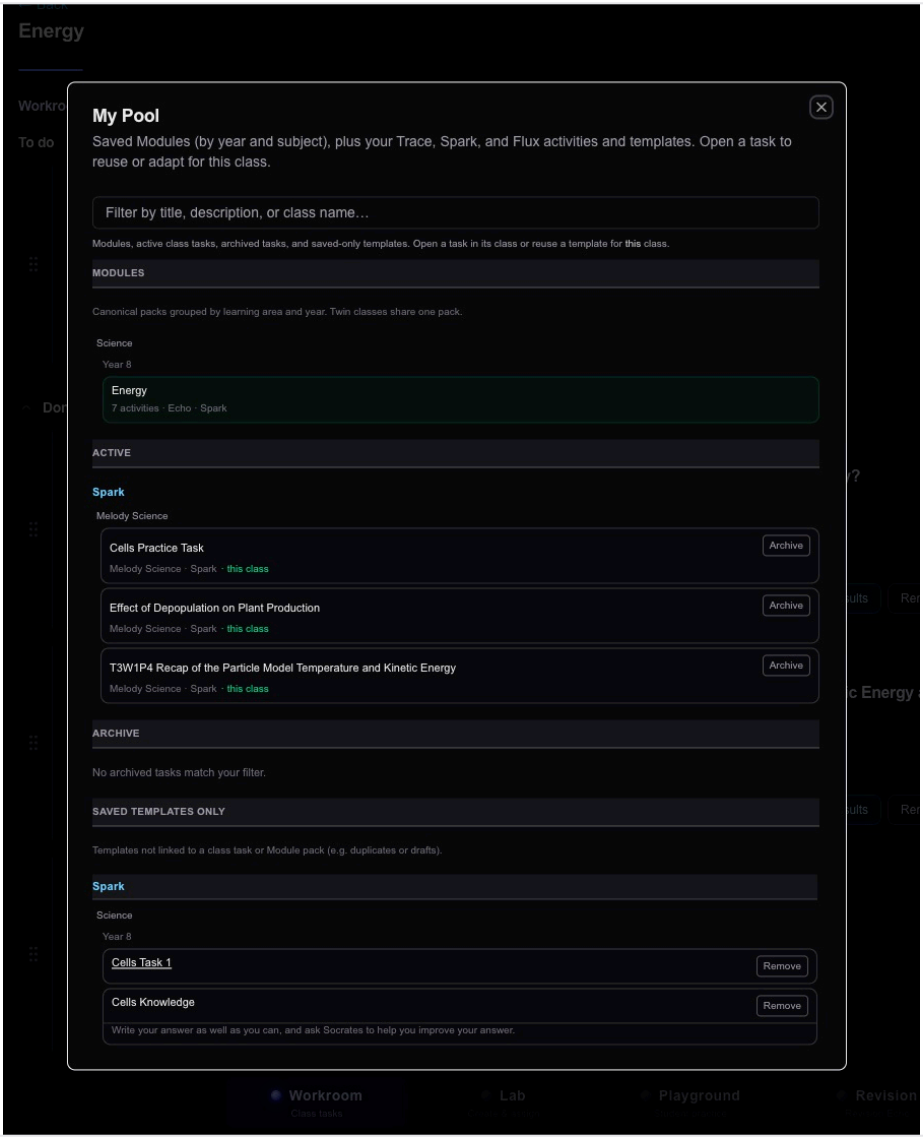
My Pool

After you have built a Module, save it to My Pool and relaunch packs or templates in other classes — your personal reuse shelf.

What My Pool is for

My Pool is your personal teacher shelf — separate from the class home. When you have a Module working well, save its activities as a reusable pack. Next time you teach the same unit, pull from My Pool instead of rebuilding from scratch.

Open My Pool from the sidebar while you are in a class. It holds Module packs you have saved and individual activity templates (Spark, Flux, Echo, and so on). If you teach twin classes with the same year and subject, the Suite can dedupe sensibly when you save.



My Pool — personal Module packs and activity templates

Saving a Module to My Pool

1. Open a Module you have already built out — with Workroom tasks assigned.
2. Choose Save module to My Pool, or Update My Pool when you have added new tasks since the last save.

3. Tick which activities to include. Items already in the pack are locked; you can add new ones.
4. Workroom cards show In My Pool or Not in My Pool yet so you always know what still needs syncing.
5. Later, in another class, create a new Module and pull from My Pool when you start a similar unit.

Packs carry year and subject labels for browsing. My Pool is for teachers only — students never see it.

Activity templates

When you create from Lab → New task, My Pool lists your saved templates for that app. Pick one to pre-fill the setup form, then assign into the Module you are working in.

- Templates stay personal unless you share them through a saved Module pack
- Archived templates drop out of the active browse list
- Oracle can help draft Spark, Flux, or Echo setup — save those drafts to My Pool when you are happy with them

Coming soon: School Pool — a school-wide shelf for sharing programmes, with Head Teachers able to publish material for classroom teachers to pick up and use.

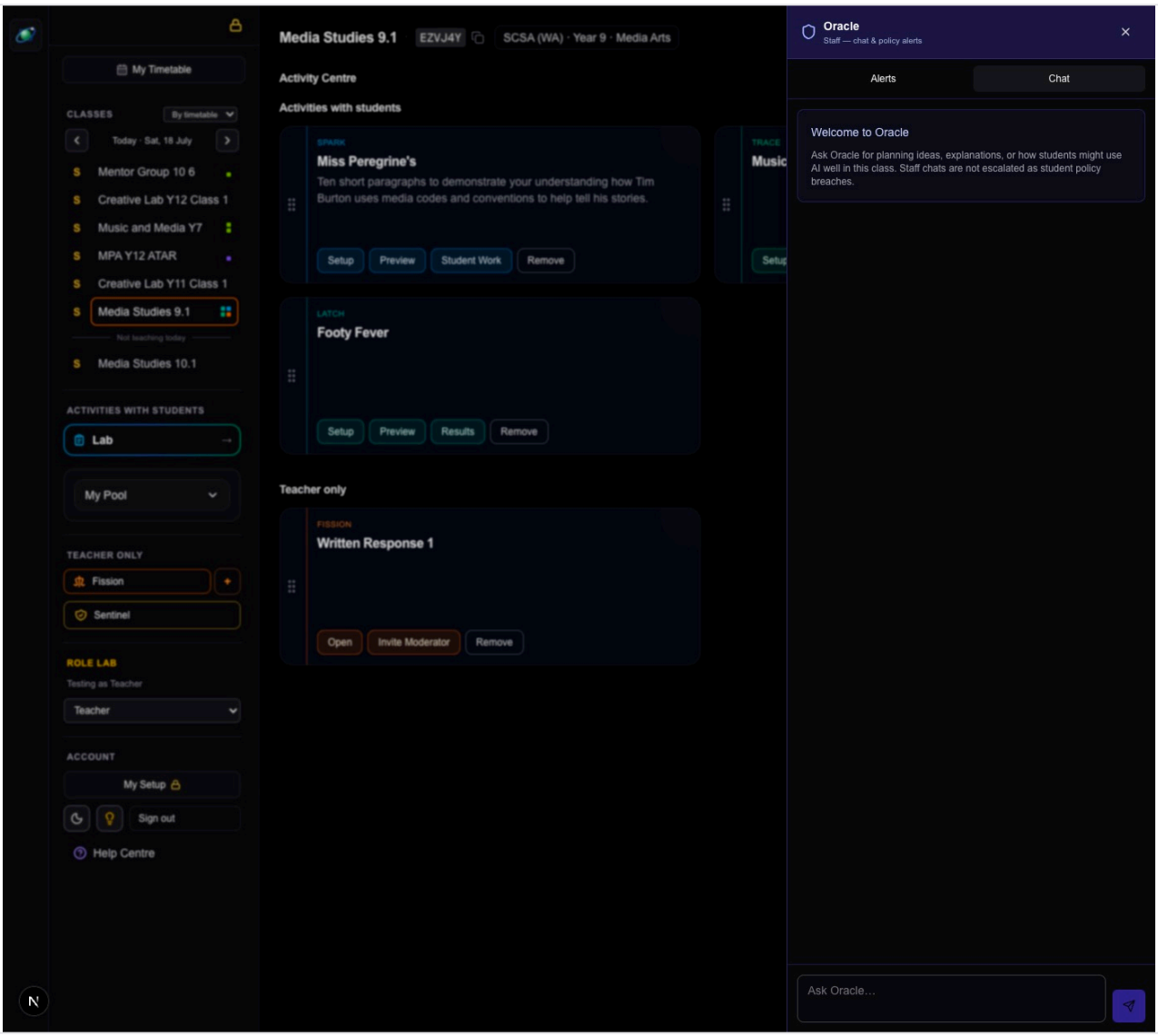
GUIDE 6 OF 17 · AI COACHES

Oracle

The class AI helper — turn it off for a class, lock it on assessments, and review policy alerts plus how much help students needed on Spark and Flux tasks.

What Oracle is

Oracle is the floating AI panel students and teachers can open during class work. It follows your school's AI use policy. It is not a Lab app, and it is not the same as Socrates — the in-task coach inside Spark, Flux, Foundry, and Echo.



Oracle panel — chat and Alerts

Role	Job
Student	Ask for learning help when Oracle is allowed. Off-topic or dishonest use gets a gentle redirect; repeated or serious misuse escalates to staff.
Teacher / staff	Use the same chat for your own questions; review Alerts when policy is breached; on Spark and Flux gradebooks, read a short summary of how much Oracle help each student needed — not a full transcript.

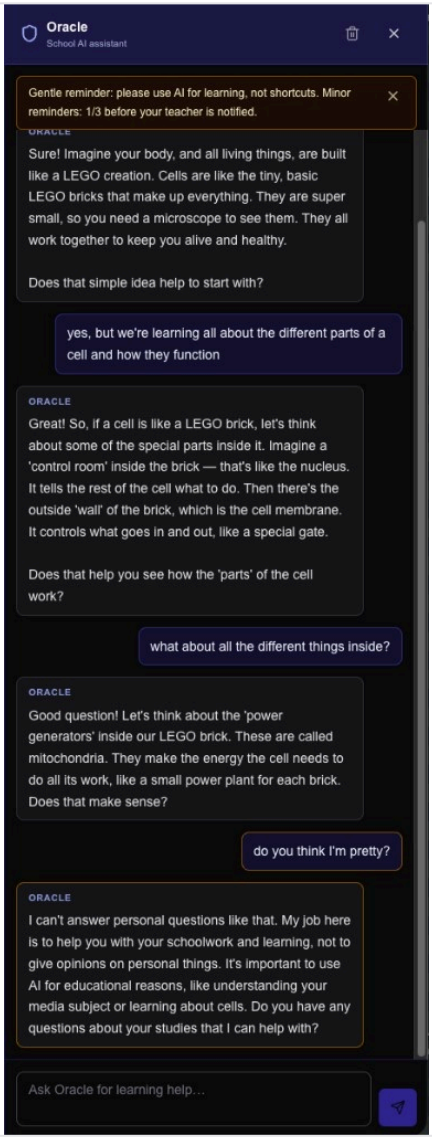
Class on/off and per-task control

You have two levels of control. First, switch Oracle off for the whole class in Suite settings — when it is off, students do not see the Oracle button in that class at all. Second, on each Spark, Flux, and Echo task, choose Practice or Proper assessment and whether to allow Oracle during that sit. Practice tasks default to Oracle on; assessment tasks lock Oracle by default.

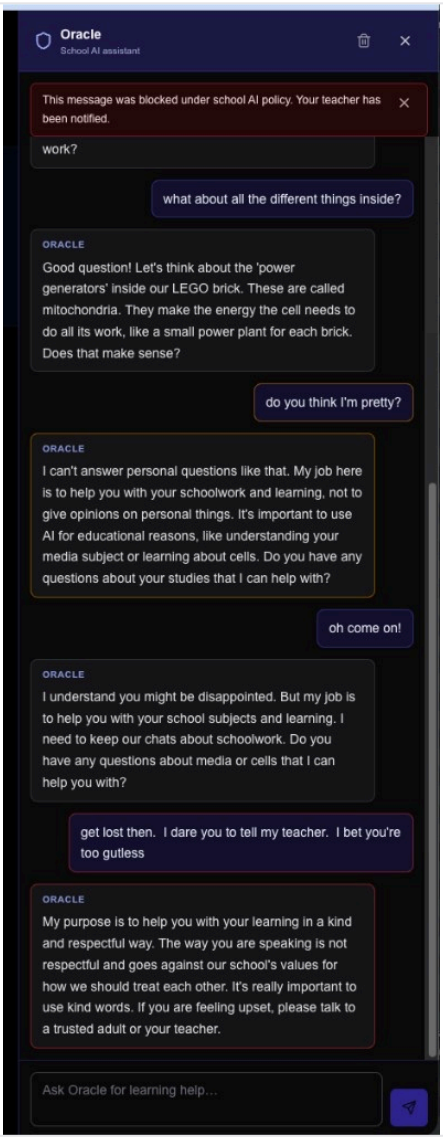
> Locking Oracle on an assessment does not remove Socrates. They are separate assistants and can both be available on the same task if you allow it.

Alerts and diagnostics

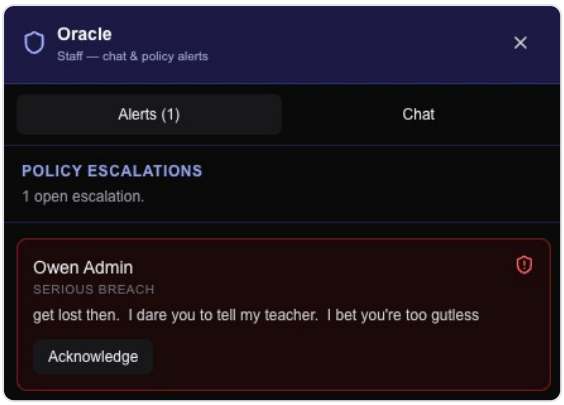
Outcome	What you see
Policy alert	Serious misuse or three minor reminders in a term escalates to Oracle → Alerts for staff to acknowledge.
Task summary (Spark / Flux)	On the student's gradebook page: how many Oracle asks they made, with sample questions — enough to see what help they needed, not a full chat log.
Everyday chat	Staff cannot browse ordinary student Oracle conversations — by design, for student privacy.



1 — Minor (1/3): redirected; teacher not notified yet



2 — Serious: blocked; teacher notified



3 — Staff Alerts: escalate → Acknowledge

Minor reminders count up through the term (1/3, 2/3, and so on). The third minor reminder, or any serious breach, lands in staff Alerts. To review alerts in Role Lab, open a class as Teacher, then Oracle → Alerts.

Oracle without a class

Teachers can open Oracle from the Suite sidebar even when no class is selected — useful for planning, policy questions, or drafting a task before you pick a Module. Class-scoped alerts and student diagnostics still require a class context.

Oracle polish on Flux

On Flux essays, Oracle can run in polish mode after Socrates and drafting — a limited number of consultation passes to tighten wording and exam readiness, separate from in-task Socrates coaching. You set Allow Oracle and polish limits at create time.

Socrates coaches structure and argument while the student writes. Oracle polish is an optional final pass — not a replacement for Socrates or for the student's own drafting.

Oracle vs Socrates

Assistant	Where	Job
Oracle	Floating panel on class pages	General learning help + school AI policy + staff alerts
Socrates	Inside Spark / Flux / Foundry / Echo tasks	In-task coaching grounded in class and syllabus context, limited by the teacher

Locking Oracle does not remove Socrates. They can both be available on the same task.

GUIDE 7 OF 17 · AI COACHES

Socrates

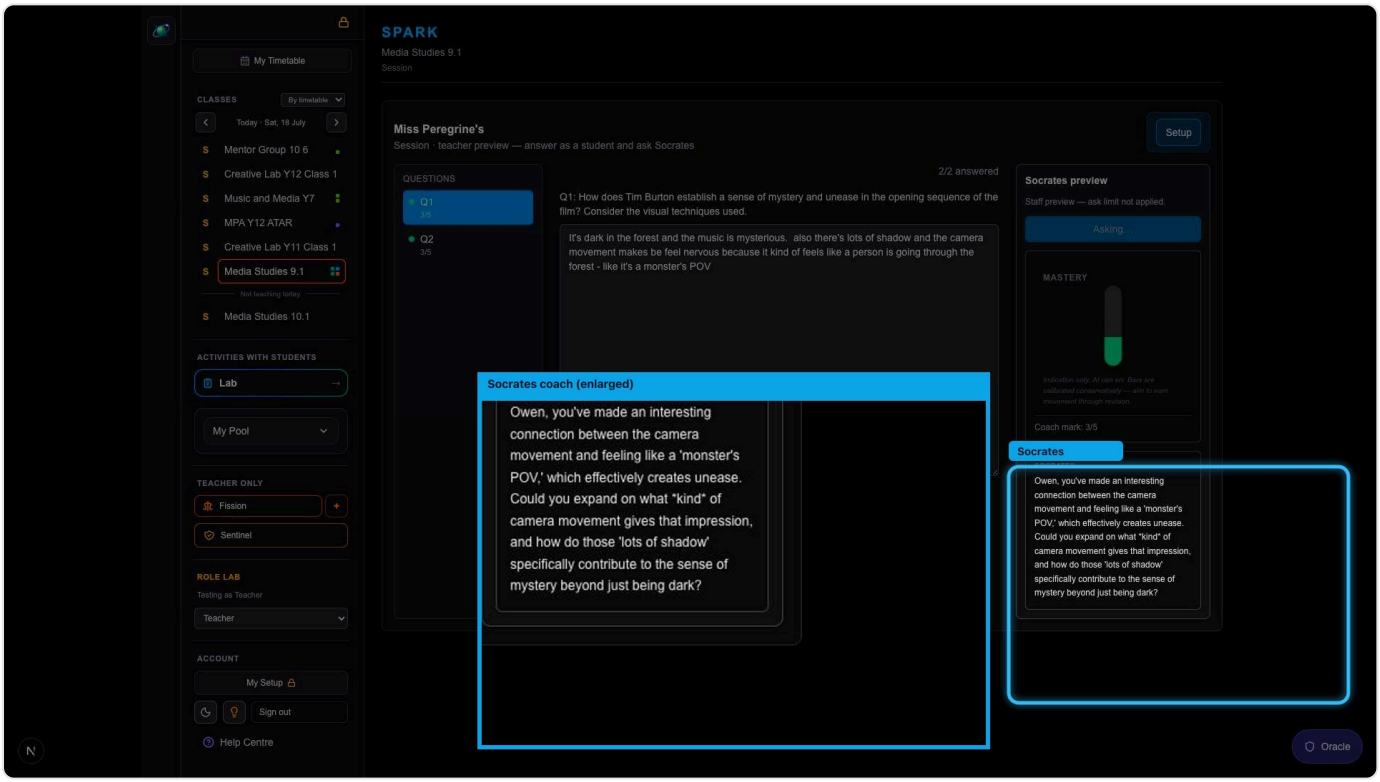
The in-task AI coach inside Spark, Flux, Foundry, and Echo — grounded in your class, syllabus, and the material you provide.

What Socrates is

> Socrates is the coach inside an assignment session. Oracle is the general class assistant in the floating panel. They do different jobs and you control them separately.

In Spark and Flux, Socrates helps students develop their own answers and structure — stretching thinking without handing over a finished paragraph or essay. In Foundry, Socrates rehearses skills you have flagged as gaps. In Echo, Socrates closes a missed revision item with one short, guided line after a retry — not a long coaching chat. You set ask limits where the app supports them.

Socrates is not a blank chatbot. It draws on your education authority syllabus (for example SCSA), the class, the student's year group, and any links or document notes you — or the student, in Playground — provide.



Socrates — in-task coach highlighted in a Spark session (enlarged callout)

Vs Oracle

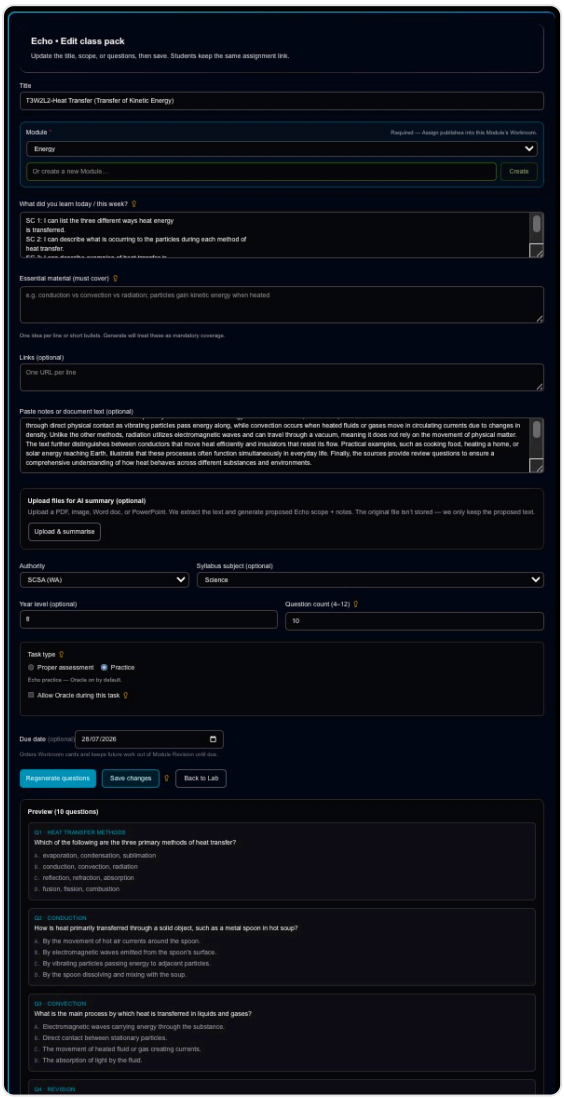
- Socrates lives inside the assignment — Spark session, Flux essay, Foundry practice, Echo retry
- Oracle lives in the class-wide floating panel for general help
- Turning Oracle off does not turn Socrates off
- Both are expected to support honest, learning-focused work — not shortcuts

GUIDE 8 OF 17 · LAB APPS — CREATE INSIDE A MODULE

Echo

Quick lesson-tied revision — multiple-choice packs from what you taught today, with a brief upskill when students miss a question.

What Echo is for



Create — lesson focus, authority, year, Practice, Allow Oracle

> Echo is for short, lesson-tied revision — usually a handful of multiple-choice questions from what you taught today. Foundry is different: longer skill-pathway diagnostics across a subject. Use Echo for quick checks; use Foundry when you want a broader gap map.

You describe the lesson focus (and optional essential material), and the Suite generates a small MCQ pack. Students sit it privately or as a class task. When they miss a question, Echo can offer a short upskill read and one retry. Assign into a Module Workroom for the whole class, or let students keep practice private in Module Playground.

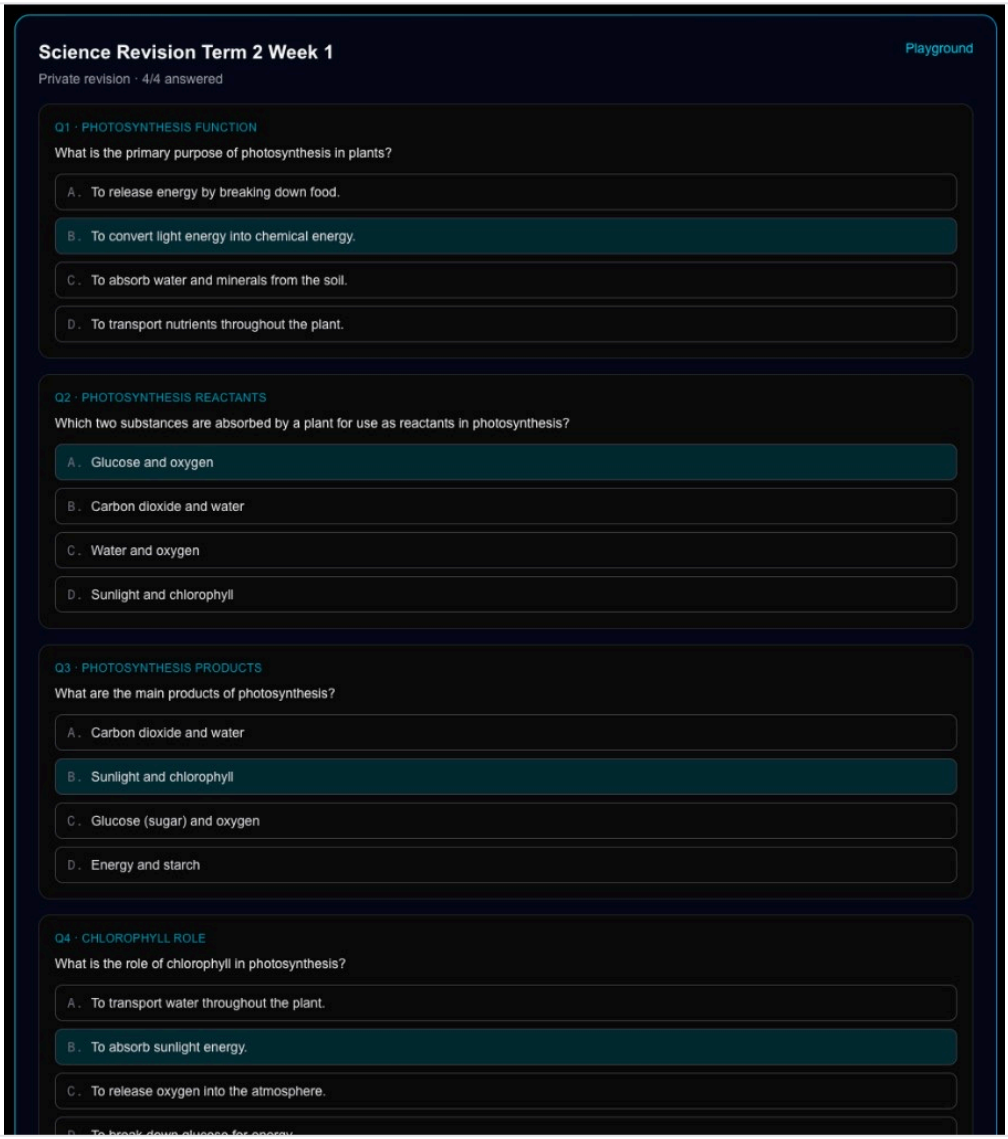
What Echo (and Socrates) draw on

Echo questions and Socrates lines are grounded in real teaching context — not a generic chat. When you draft a pack, the generator uses:

- Your education authority syllabus (for example SCSA) when you pick a subject
- The class and student year level you are creating in
- What was learned today — your lesson notes
- Essential material — the must-know facts, terms, or steps Echo should test (separate from broader lesson notes)
- Any extra links or pasted document notes you provide
- In Playground, resources the student adds for that week

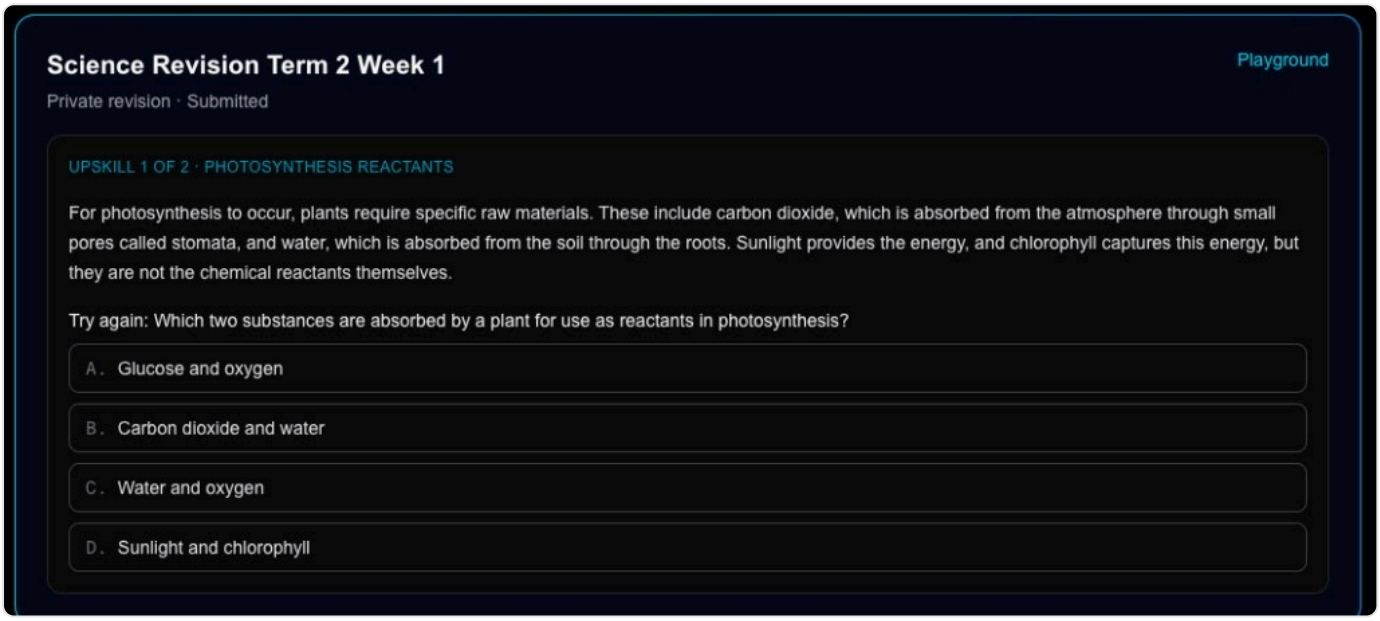
Typical flow

1. Open your Module → Lab → Assign Echo pack (or use Student Lab / Playground for private practice).
2. Set lesson focus, essential material, links and notes, authority subject, year level, Practice or assessment, Allow Oracle, and an optional due date.
3. Publish to the Module Workroom for the class, or keep the pack private in Module Playground.
4. Students sit the MCQs. On a miss, they may see diagnostics, a short upskill paragraph, and one retry.
5. On a second miss, Echo underlines the answer span, offers one Socrates line, then asks a quick self-check — do you feel you have got it? Ready to move on? — without opening a long chat.



Sit the pack — short private revision MCQs

Results — On track vs Gap, then optional Upskill



Upskill — short read, then one retry on the gap

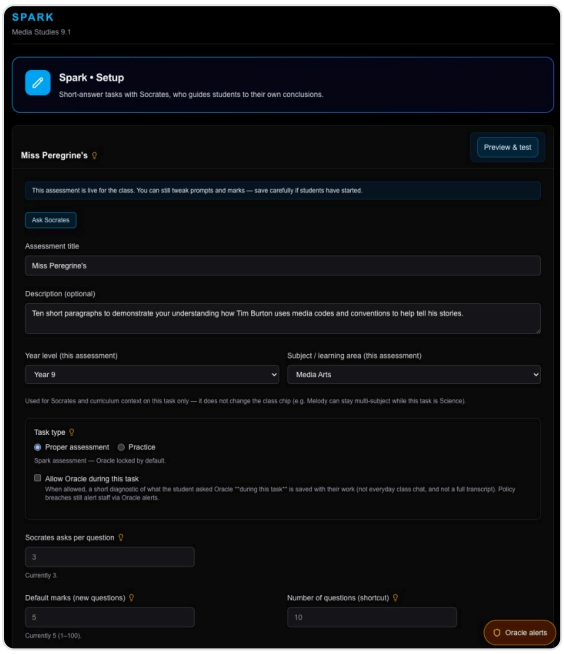
Socrates in Echo is deliberately brief: one guided line after a second miss, then the student moves on. Spark and Flux offer a fuller coaching conversation; Echo is meant to close a gap quickly and return to the pack.

GUIDE 9 OF 17 · LAB APPS — CREATE INSIDE A MODULE

Spark

Short-answer writing with Socrates coaching — depth in a small space, with Practice or assessment and optional Oracle summaries on the gradebook.

What Spark is for

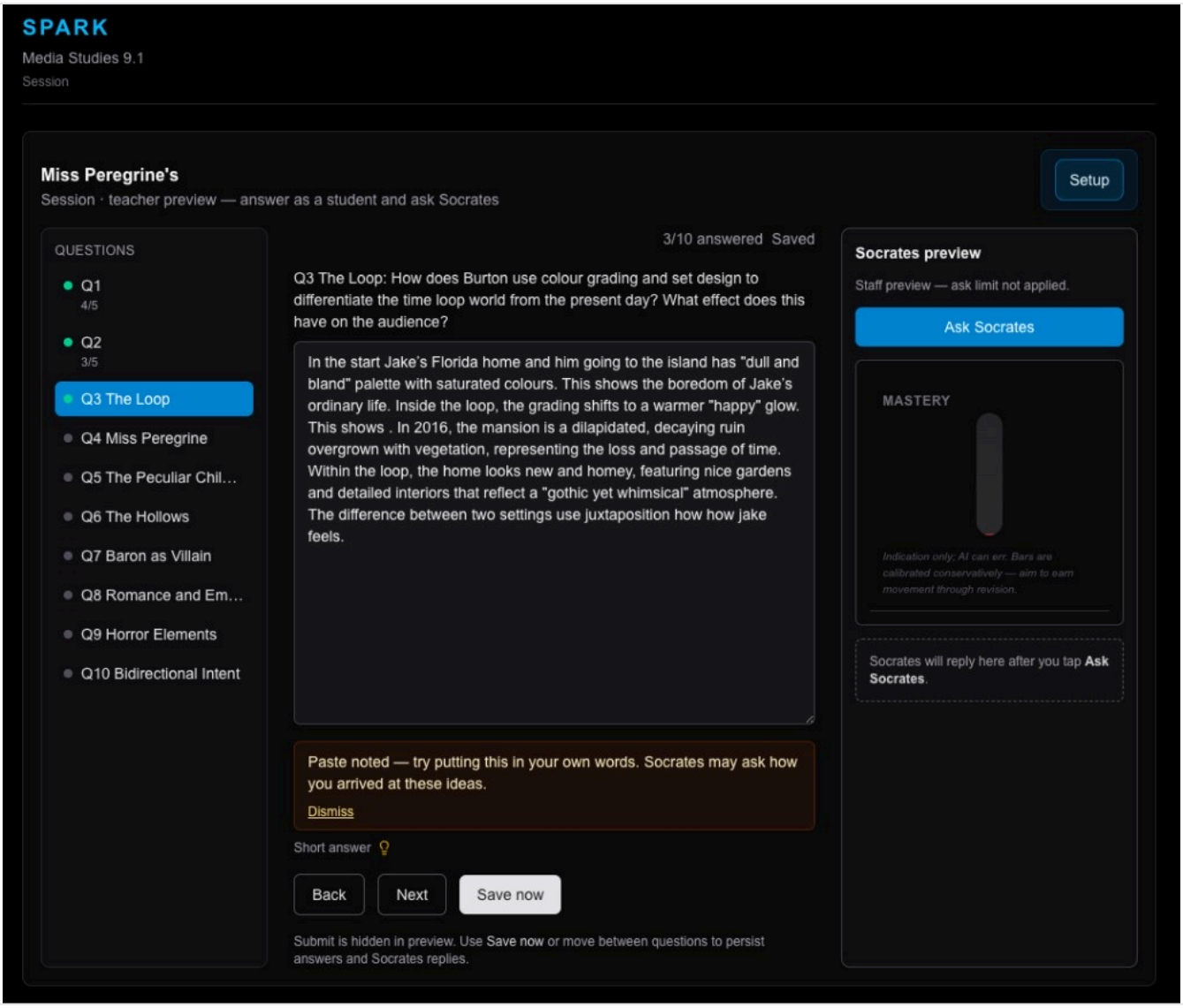


Setup — title, Practice or assessment, Allow Oracle, Socrates asks, marks, question count

> Spark is for short written answers with Socrates beside each question. Choose Spark when you want depth in a small space; choose Flux when you need a full essay with structure and a target word count. Oracle can sit alongside either when you allow it.

You write one or more focused questions, set how Socrates should coach, and choose Practice or assessment. Students answer in the session; Socrates stretches their thinking without writing the paragraph for them. Spark suits lesson exits, concept checks, and exam-style short responses.

Student session and paste authenticity



Session — paste noted nudge; Socrates may probe how they arrived at the ideas

If a student pastes a substantial block of text into an answer (about forty characters or more), Spark shows a gentle nudge: “Paste noted — try putting this in your own words. Socrates may ask how you arrived at these ideas.” That is not an accusation — it signals that ownership matters. The paste is recorded on that question for you and steers Socrates toward at least one “how did you arrive at this?” question, without mentioning paste detection to the student.

- Students can dismiss the nudge and keep working
- You can see a paste signal on that question in the gradebook
- Flux uses the same pattern on long-form essays

> The goal is ownership of ideas, not a plagiarism tribunal.

How you use it

1. Open your Module → Lab → Create Spark activity.
2. Write prompts and what you are looking for; set Practice or assessment, Oracle, Socrates asks per question, marks, and an optional due date.
3. Publish to the Module Workroom, or to Playground for private practice patterns.

4. Students sit the task, use Socrates within your limits, and submit.
5. Open the gradebook or results; if Oracle was allowed, read the short summary of what help each student needed.

For name-and-identify questions, Socrates can confirm a correct match without forcing an essay. For explain-and-extend items, coaching pushes for evidence and depth.

GUIDE 10 OF 17 · LAB APPS — CREATE INSIDE A MODULE

Flux

Long-form writing with a target word count — Socrates expands or truncates toward high-quality exam answers using structure, Bloom's, and They Say / I Say.

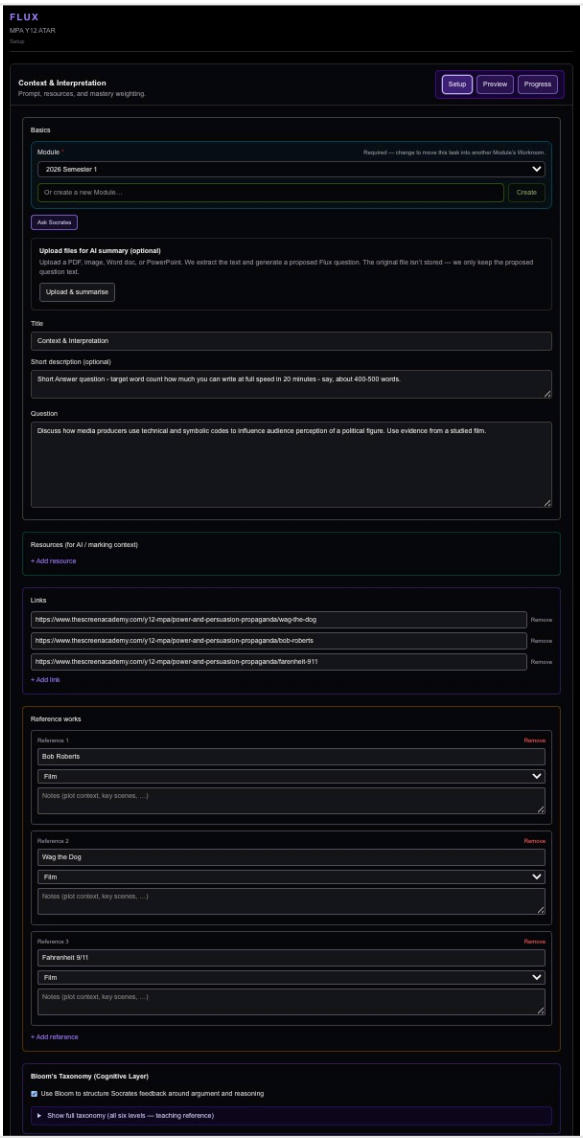
What Flux is for

Flux is for sustained writing. Use Spark when you want short answers; use Flux when structure, argument, and exam-length stamina matter.

The target word count is central to Flux. It helps two kinds of student at once: those who need to expand and fill a whole exam paper with enough knowledge and detail, and those who need to distill what they know into the time allowed — succinct, high-quality answers rather than sprawl.

A simple way to set the target: write flat out for five minutes, count the words, multiply by ten. That estimates how much they can write in about fifty minutes (for example, 120 words × 10 → ~1200 words). That figure becomes their target word count in Flux — a realistic personal ceiling for exam-length practice, not an arbitrary quota.

Expand to fill the paper when they under-write; distill to fit the clock when they over-write. Same tool, same target — different coaching pressure depending on the student.

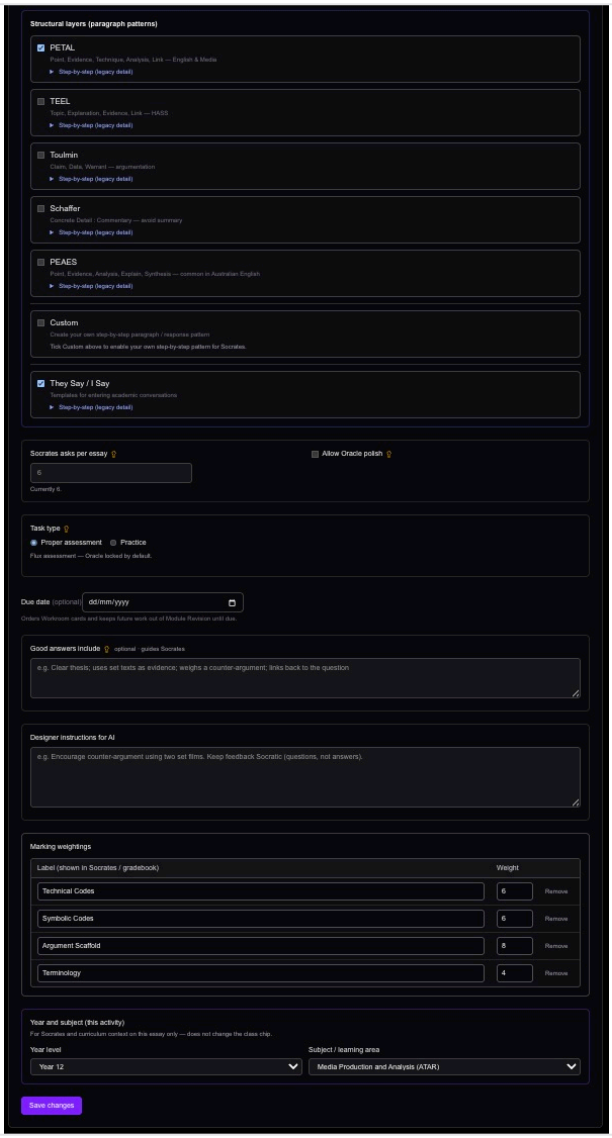


Setup — title, question, and links for Socrates context

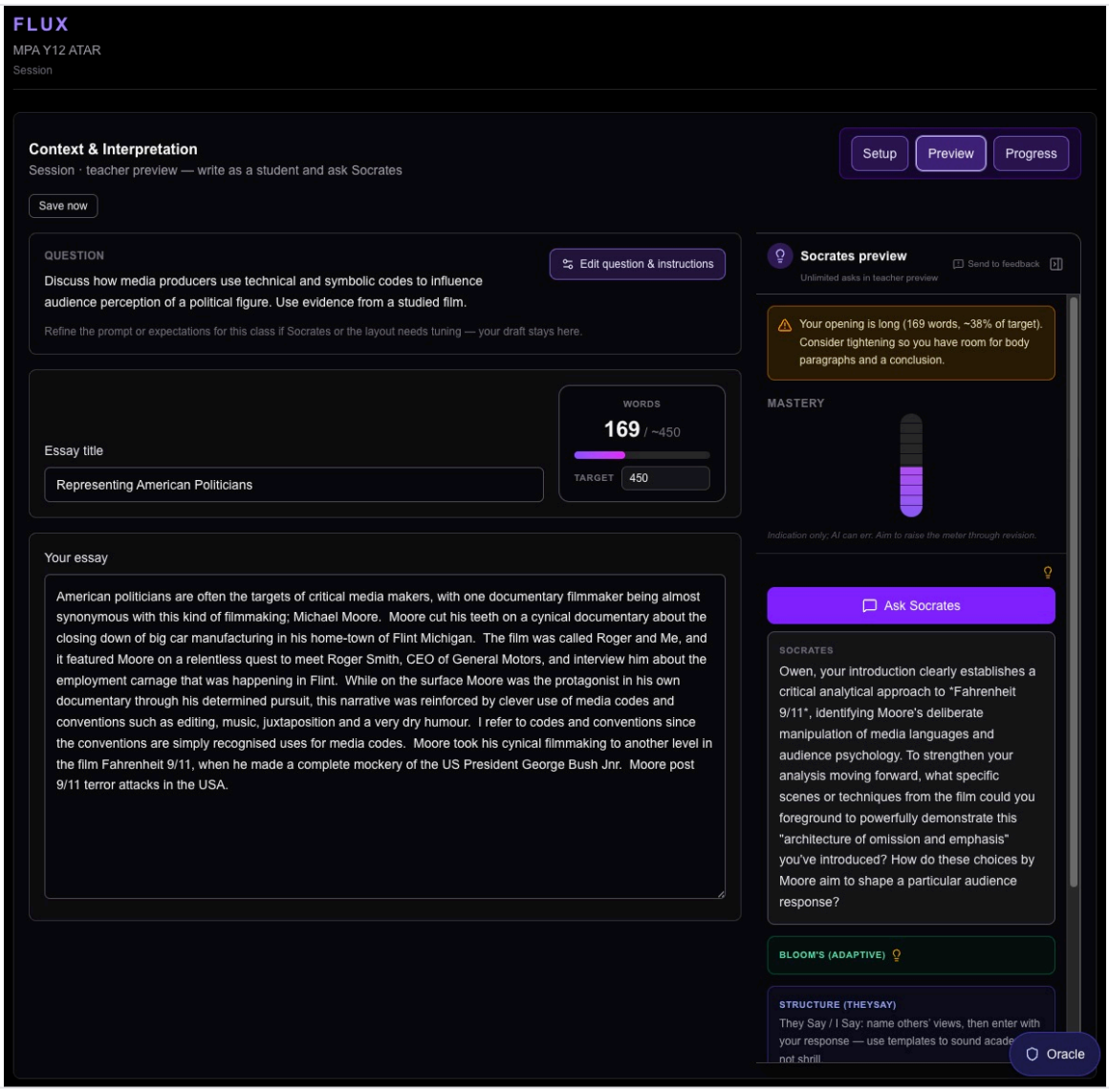
Socrates is how Flux turns word-count practice into writing training. When a student needs to grow the piece, Socrates coaches expansion; when they are over length or rambling, Socrates helps them truncate and tighten — always toward the target, not just “write more” or “write less.”

Coaching is anchored in designated structural requirements for the task, and in established frameworks for academic quality — including Bloom’s taxonomy (depth of thinking) and They Say / I Say (voice, response, and argument moves). The aim is the highest-quality answer that fits the time and the target word count.

Flux + Socrates = exam-length stamina with structure, thinking level, and argument craft — not a word counter alone.



Setup — PETAL / They Say I Say, and Socrates asks per essay



Session (teacher preview) — target word count, Mastery Meter, and Socrates coaching on structure

Create from Module → Lab → Flux create. Same Practice / assessment and Allow Oracle pattern as Spark. Ground Socrates with studied texts (links and optional reference works), turn on Bloom's for argument depth, pick paragraph patterns (e.g. PETAL) and They Say / I Say, then set Socrates asks per essay.

> Large pastes on a Flux essay get the same authenticity treatment as Spark: a student nudge, Socrates ownership questions, and a staff paste signal — without accusing cheating.

How you use it

1. Open your Module → Lab → Create Flux activity.
2. Set brief, structural expectations, task type, Oracle (including polish limits on Flux), coaching, and optional due date.
3. Publish → Module Workroom.
4. Students set or use a target word count (often from a five-minute flat-out sample × 10), draft with Socrates expanding or truncating toward that target, and submit.
5. Review on the gradebook.

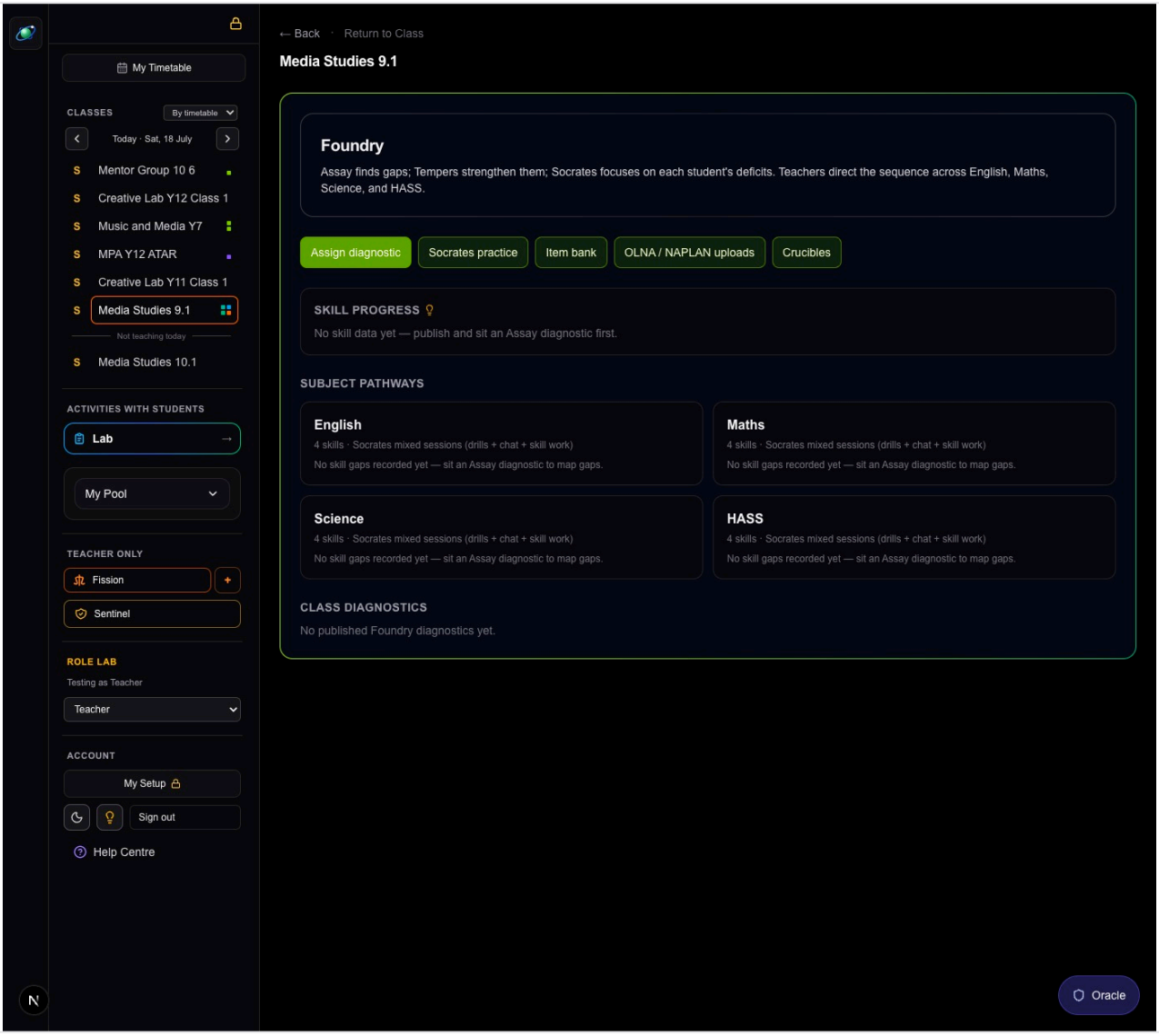
GUIDE 11 OF 17 · LAB APPS — CREATE INSIDE A MODULE

Foundry

Formative diagnostics that find skill gaps, then coach and re-check — useful for OLNA preparation, not an official SCSA result.

What Foundry is for

Foundry helps you find what students are missing, teach and coach those gaps, then re-check with short follow-ups — without running another high-stakes paper every week.



Foundry home — pathways and diagnostics

Foundry runs in three modes. Assay is the first full diagnostic (~32 items) when you start a subject. Temper is a shorter follow-up (4–12 items) while you are teaching known gaps. Quench is a mid-term or pre-window check (8–16 items). The usual rhythm is Assay → teach and Socrates practice → Temper → teach again → optional Quench.

How you use it

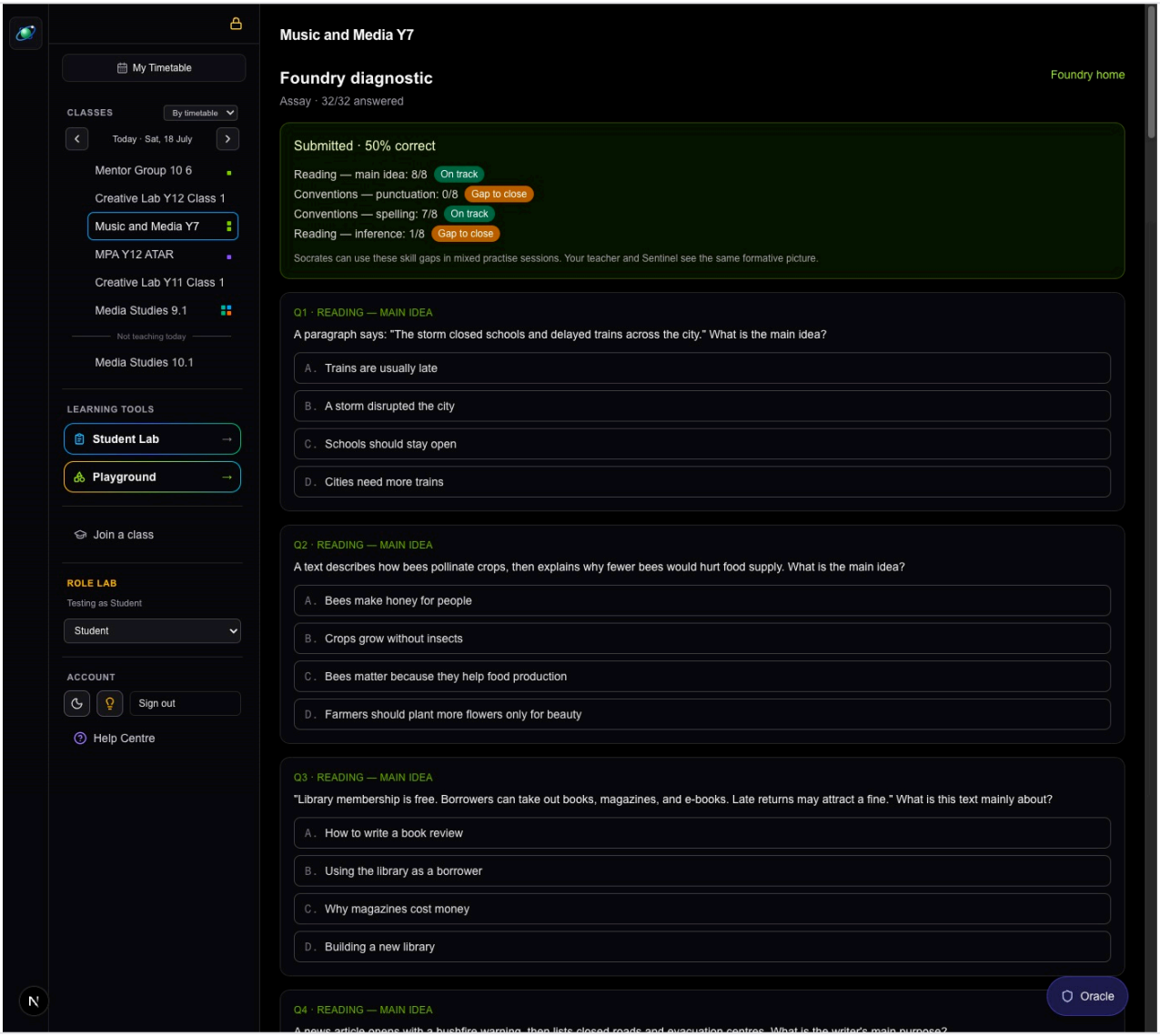
1. Open your Module → Lab → Assign Foundry diagnostic.
2. Choose Assay, Temper, or Quench; set subject, year band, and an optional due date, then publish.

3. Students sit the diagnostic and submit. Results repeat at the foot of the sit with suggested next steps.
4. Open Results for class gaps and plan your next teach or Temper follow-up.
5. Students practise with Socrates on concrete misses — coaching is teach-first for struggling learners.
6. Later, assign a shorter Temper on the same subject. Done Workroom cards show compact practice needs at a glance.

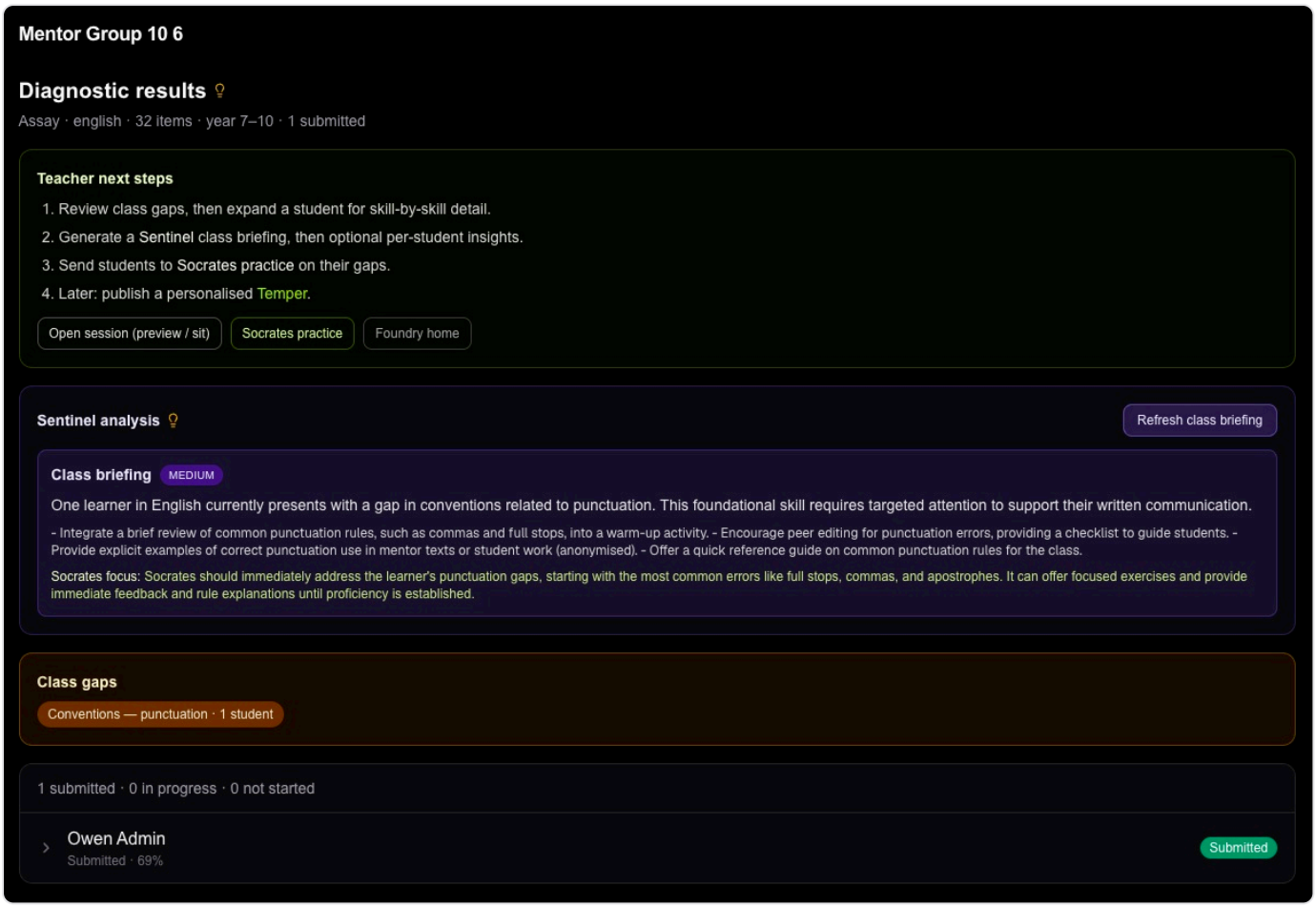
A skill gap means mastery under about 60% on that skill in the diagnostic — a planning flag, not a pass/fail grade. Items are personalised by year band, subject, and prior responses. They are original formative practice for OLNA preparation: not official SCSA papers and not an official Category result.

Key screens

Foundry home shows pathways and your list of diagnostics. Create assigns a new Assay, Temper, or Quench. Students sit from Session (you can Preview first). Results shows scores and class gaps. Practice opens Socrates on current gaps.



Student session after submit — score, On track / Gap to close, then items



Teacher Results — class gaps and Socrates practice

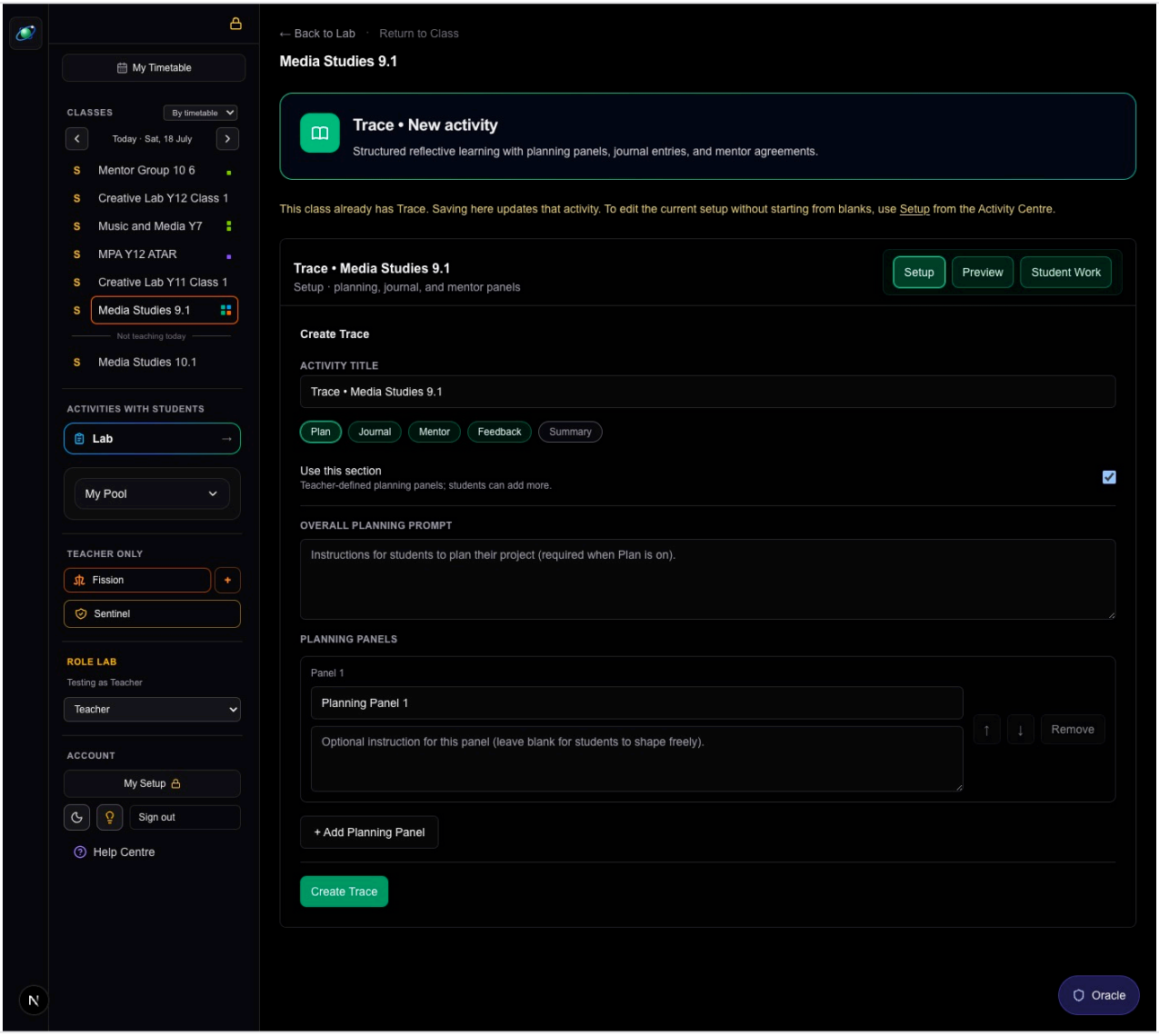
GUIDE 12 OF 17 · LAB APPS — CREATE INSIDE A MODULE

Trace

Planning and reflective journalling — entries, planning panels, and mentor agreements that build a student's thinking trail over time.

What Trace is for

Trace is for planning and reflective journalling — student artefacts over time, not a one-off quiz. It sits in the Lab family as a process and reflection tool.



Trace — planning and reflective journalling

Use Trace when you want students to plan, journal, and revisit thinking across a unit — not only to submit a finished answer. Entries become a trail you can mentor against: what they intended, what they tried, and what they learned.

- Planning panels — structure the work before they write or make
- Journal entries — reflective writing tied to the class activity
- Mentor agreements — shared expectations between teacher and student where you set them up

How you use it

1. Open your Module → Lab → Create Trace activity.
2. Configure prompts, planning structure, and any mentor agreements, then publish.
3. Students open the activity from the Module Workroom.
4. Review progress from class Trace views and Workroom — coach the thinking trail, not only the final product.

Trace pairs well with Spark or Flux: students plan and reflect in Trace, then produce the assessed response in a writing app.

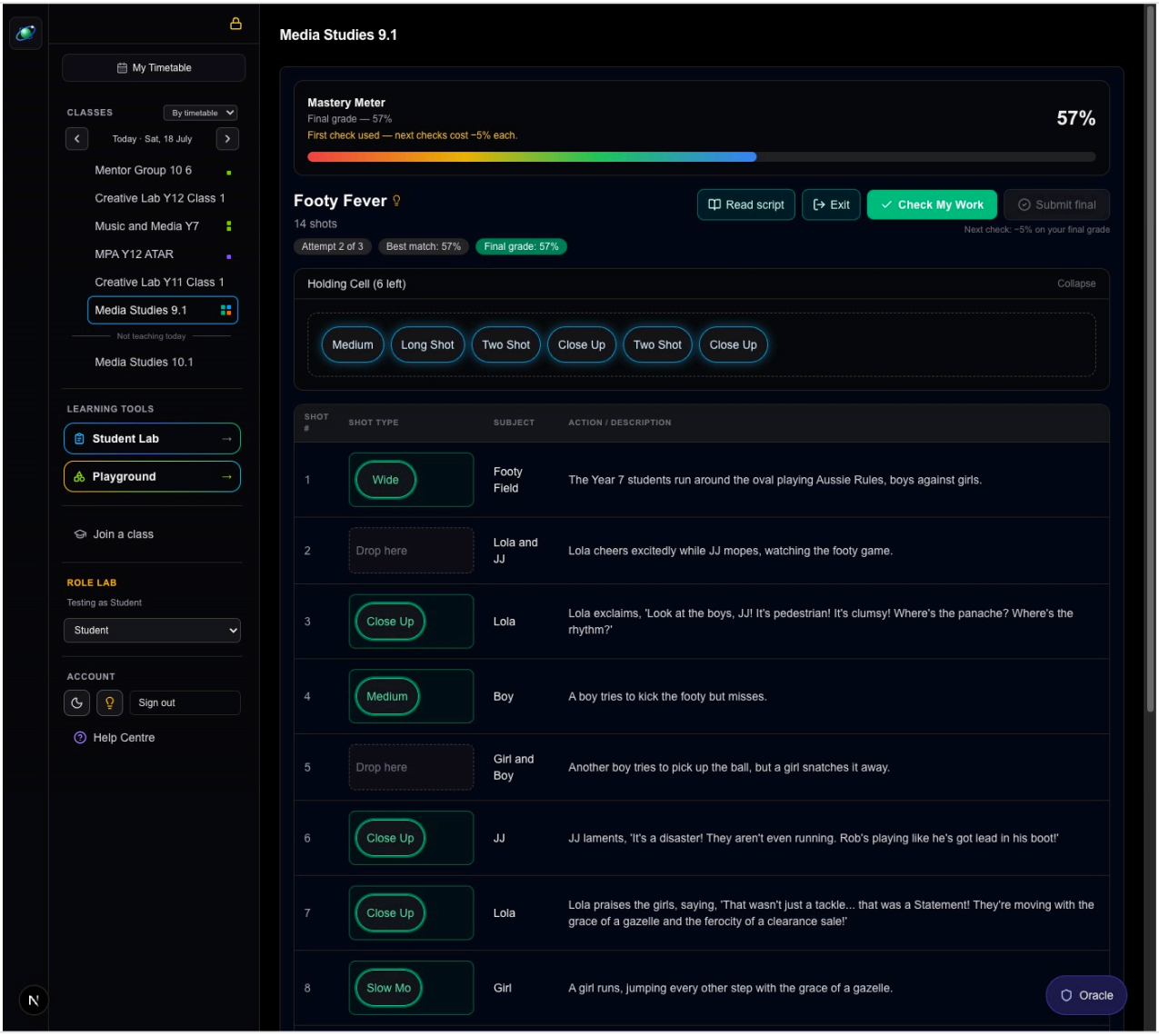
GUIDE 13 OF 17 · LAB APPS — CREATE INSIDE A MODULE

Latch

Match-up and shot-listing — drag chips into rows. Built for Media Arts storyboards, and any categorise-and-match task.

What Latch is for

Latch is categorise-and-match: students drag chips from a Holding Cell into the right rows — shot types into a storyboard, labels into categories, or any “put this with that” task.



Student session — Holding Cell chips into shot rows; Check My Work unlocks the Mastery Meter

In Media Arts, Latch is often a shot list: students match Wide, Close Up, Two Shot, Slow Mo, and so on to script beats (subject and action). The same pattern works for any subject where students must classify and place items correctly.

How a student session works

- Holding Cell — unused chips wait here until dragged onto a row
- Rows — shot #, shot type drop zone, subject, action / description
- Check My Work — first check free; later checks may cost a grade penalty (shown on the Mastery Meter)

- Submit final — when the match-up is ready
- Read script — open the source script or brief if you attached one

> The Mastery Meter unlocks after Check My Work so students see colour-coded match quality — then they can revise chips before a final submit.

How you set it up

1. Open your Module → Lab → Create Latch activity (or generate rows with AI where available).
2. Build the chip set and rows; publish to the Module Workroom.
3. Students open the session, place chips, check work, and submit.
4. Review results from Workroom / Latch results views.

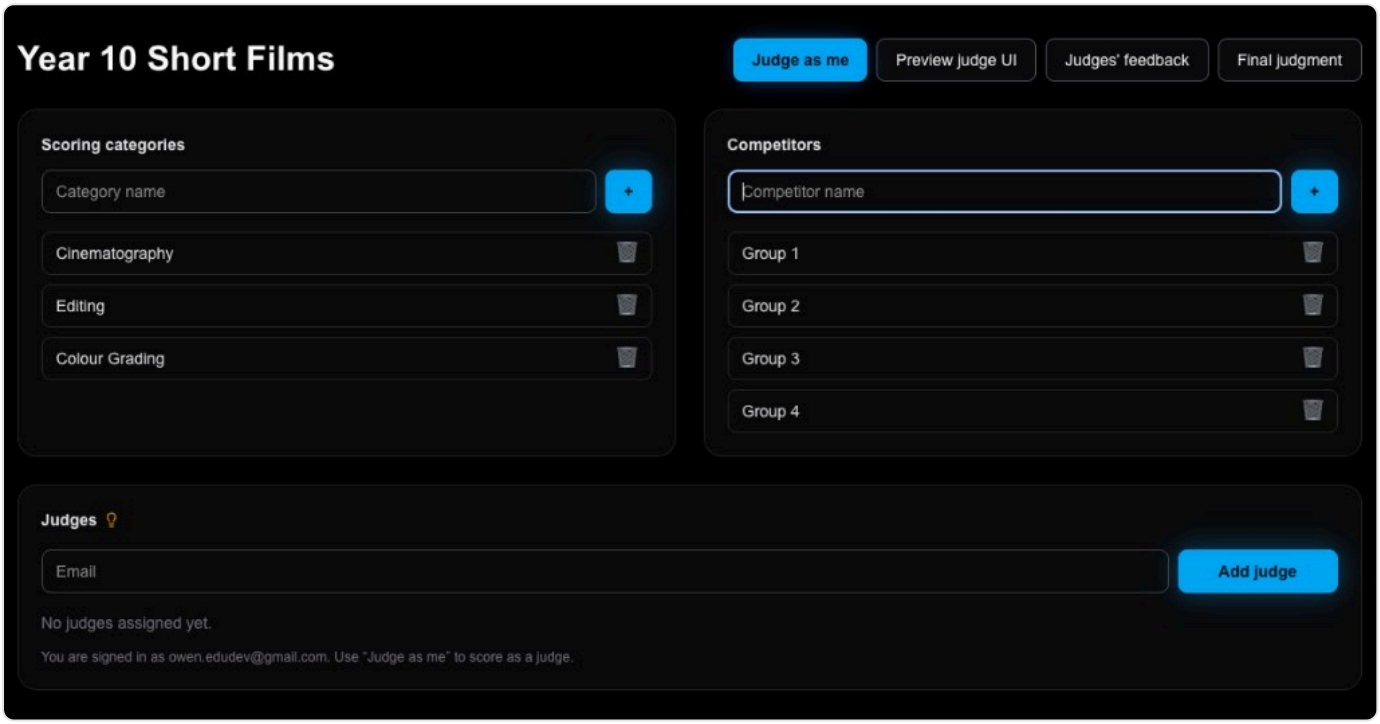
Jury

Live judging with private scores and a projected leaderboard — battle of the bands, film screenings, peer judging, or any scored showcase.

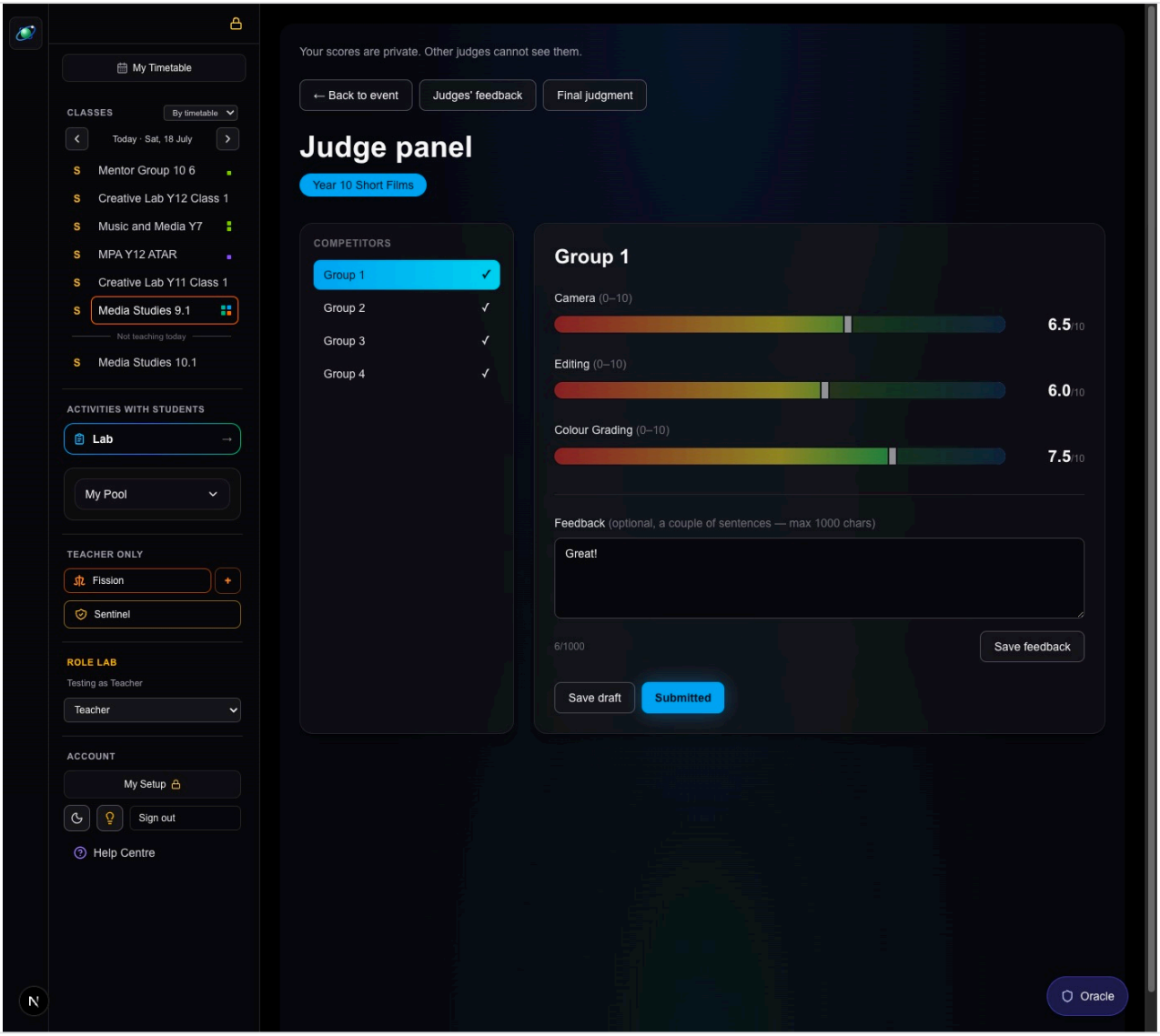
What Jury is for

Jury runs live performance or screening events: judges score privately, then you can publish and project a leaderboard for the audience.

Use it for Year 10 short films, battle of the bands, pitch nights, or peer critique with clear criteria. Scores stay private to each judge until you are ready to share; feedback can be collected separately from the final ranking.



Event setup — scoring categories, competitors, and judges



Judge panel — private scores, criteria sliders, optional feedback

Teacher, judges, and audience

Role	What they do
Teacher / host	Create the event, set categories and competitors, add judges, run the session, open Judges' feedback and Final judgment, project the live board
Judge	Open the judge panel, score each competitor on your criteria (sliders), optionally leave short feedback, submit — other judges cannot see your scores
Audience	Watch the projected leaderboard when you publish results for the room

Use Judge as me to score yourself as a judge, or Preview judge UI to see the student/guest judge experience before the event.

How you run an event

1. Open your Module → Lab → Create Jury activity; name the event.
2. Add scoring categories (e.g. Cinematography, Editing, Colour grading) and competitors (groups or acts).
3. Add judges by email, or use Judge as me for a quick staff score.
4. During the event: judges score privately; you monitor the session.

5. Open Judges’ feedback and Final judgment; project the leaderboard when ready.

Jury also closes the Arcade term loop: after students build games, the class can judge the best versions in Jury, then publish the winner into Playground for ongoing play.

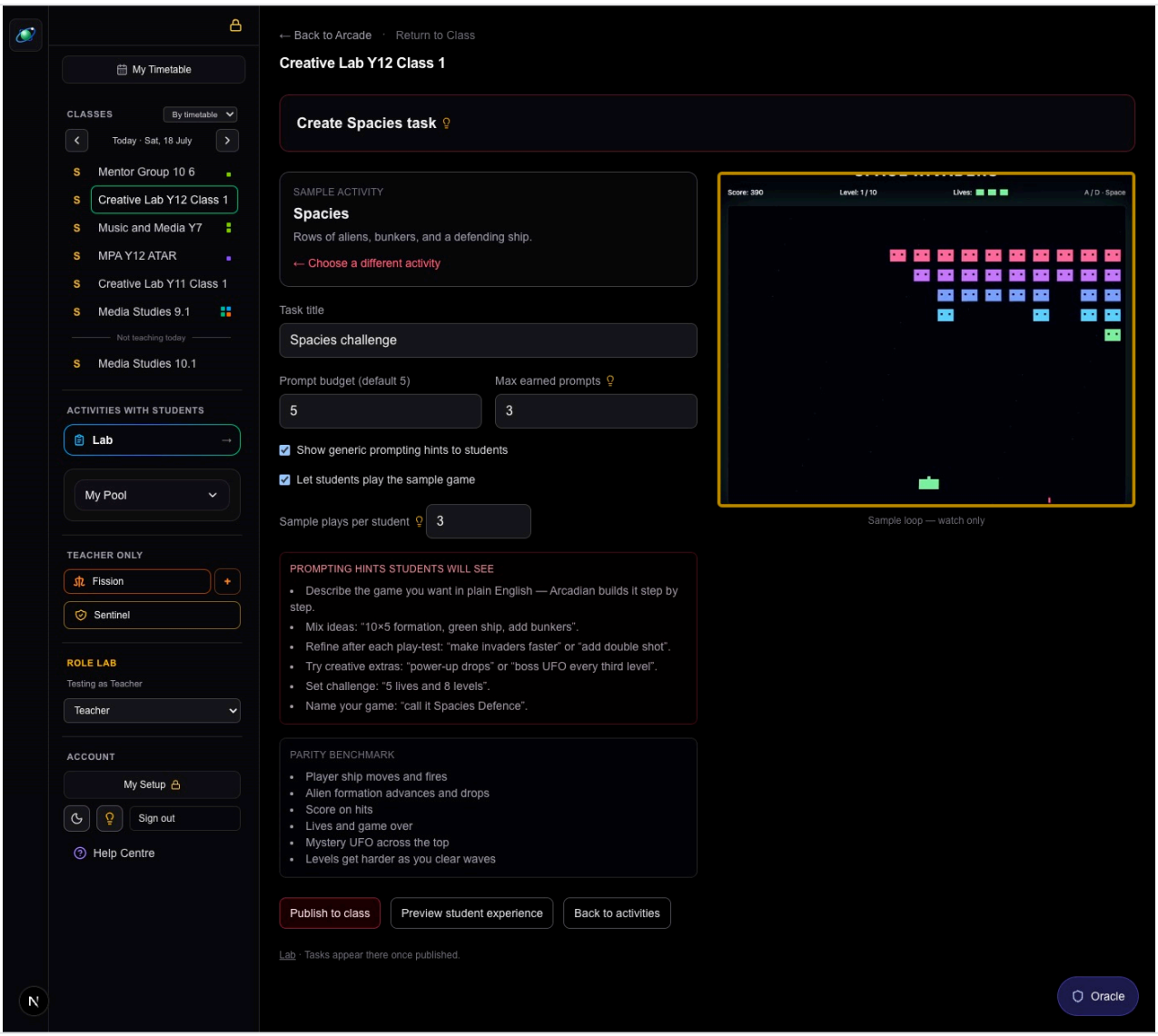
GUIDE 15 OF 17 · LAB APPS — CREATE INSIDE A MODULE

Arcade

A fun way to practise the human–AI–human cycle — creative and critical thinking through build, judge, play, and improve.

What Arcade is for

Arcade helps students practise the highly creative and critical thinking in the human–AI–human cycle — analysis, evaluation, decision-making, then round again. Fun on the surface; deep process underneath. That cycle sits in the philosophy of Isotopic as a whole.



Create Spacies — prompt budget, parity benchmark, and sample loop preview

Students study a sample loop, work with Arcadian (the Arcade AI), decide what to keep or change, and publish games for class play. Each pass through the loop is analysis → evaluation → decision → try again.

A great term pattern

Teachers can set an Arcade challenge at the start of term. The class judges the best versions with Jury. The winner is published into Playground so the class can play it from time to time for the rest of the term. Top ten scores are recorded, and a prize can be given at the end of term.

```
Start of term: set Arcade challenge
→ Students build (human–AI–human loop)
→ Jury: class judges the best version
→ Winner published in Playground for ongoing play
→ Top ten scores recorded → end-of-term prize
```

1. Open your Module → Lab → Choose Arcade activity and set the term challenge.
2. Students build and iterate with Arcadian.
3. Run Jury to judge the strongest versions.
4. Publish the winning game into Playground for the class to play across the term.
5. Track top scores; celebrate with an end-of-term prize.

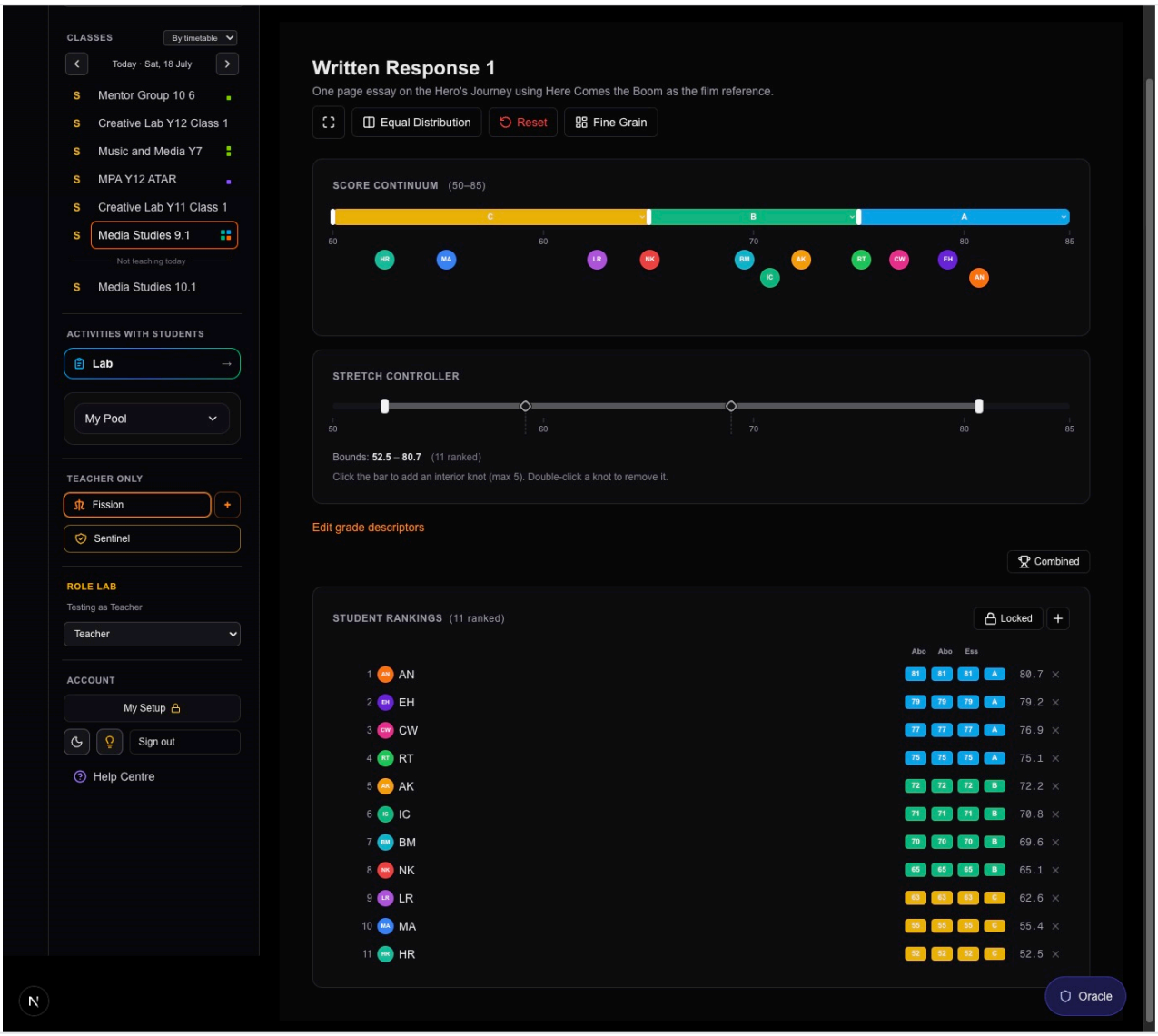
GUIDE 16 OF 17 · TEACHER TOOLS

Fission

A huge time-saver for moderation — rank, stretch, and lock grades with professional judgment. Under Teacher only; not a Lab create app.

What Fission is for

Fission is a huge time-saver for moderation. You keep relative order and professional judgment, while the continuum and stretch tools turn a stack of scripts into a defensible ranked set far faster than re-marking by hand.



Fission — score continuum, stretch controller, and student rankings

Open Fission from Teacher only in the sidebar. You work a task (for example a written response) on a score continuum: place students relatively, then use the stretch controller to set bounds and knots so letter grades and finals reflect your standards. Rankings update as you adjust — without starting again from a blank spreadsheet.

- Score continuum — visual placement of students across grade bands (e.g. C / B / A)
- Stretch controller — click to add interior knots; double-click to remove; set bounds for the ranked set
- Student rankings — combined or aspect scores, letter grade, final score; lock when ready

- Equal distribution / Fine grain / Reset — tools to reshape the spread without losing relative order
- Edit grade descriptors — align band language with your assessment

Why moderation is faster

Traditional moderation often means re-reading, re-totaling, and arguing about cut-offs late at night. Fission keeps the hard part — your sense of who is stronger than whom — and automates the tedious part: mapping that order onto marks, bands, and finals as you stretch or fine-grain the scale.

Equal distribution and fine grain let you explore fair spreads without losing relative rank. Reset and lock give you a clean path from draft judgment to a settled result you can stand behind with colleagues or parents.

> Judgment stays yours. Fission saves the hours spent recalculating and reshuffling so moderation meetings can focus on standards, not arithmetic.

Vs Lab and Sentinel

Tool	Job
Lab	Create student learning activities
Fission	Mark and moderate student work with continuum judgment — fast
Sentinel	Learning support, pastoral care, behaviour, and accommodation records (staff documentation)

> Spark and Flux gradebooks hold task scores and coaching history. Fission is where you moderate and stretch a cohort's ranked results into defensible finals.

How you use it

1. Teacher only → Fission (or open a Fission task from the class).
2. Select the assessment / response set you are moderating.
3. Order students on the continuum; adjust stretch knots and bounds.
4. Review rankings and letter grades; edit descriptors if needed.
5. Lock when the set is ready for reporting.

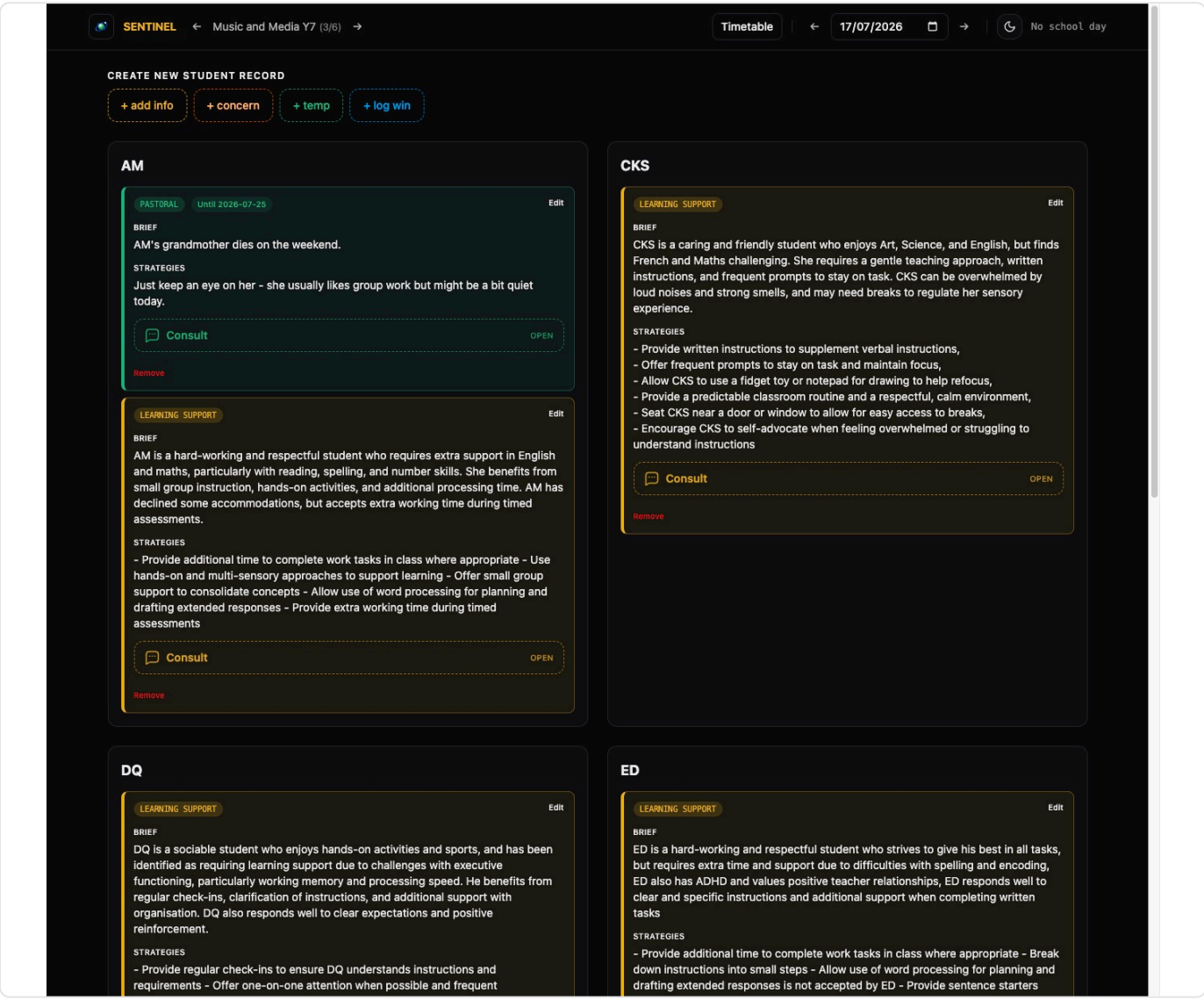
GUIDE 17 OF 17 · TEACHER TOOLS

Sentinel

Capture learning support, pastoral care, and accommodations in the moment — for a better classroom experience, compliant records, and audit-ready oversight for Sentinel Administrators.

What Sentinel is for

Sentinel is where teachers capture what they are doing for students in the moment — learning support, accommodations, pastoral care, and behaviour — so the record is authentic, not rebuilt from memory later.



Sentinel — class briefing with standing profiles, temps, and strategies

The whole Sentinel process is designed to make the student–teacher experience more authentic and meaningful. Teachers see the right brief and strategies on the day; students get consistent care in a better learning environment. When you log an accommodation as you deliver it, that log is immediately tied to what the student is entitled to and documented in a compliant way — not a Sunday-night reconstruction of what you think you might have done.

Open Sentinel from Teacher only in the sidebar, or tap the S chip on a class or timetable row. Sentinel opens in its own app with your class selected. Students never see Sentinel notes; paraphrased briefs can inform other Isotopic tools behind the scenes where that is designed in.

Why immediacy matters

Instant record-keeping saves teachers a great deal of work — and that is a direct money saver for the school. You log brief and strategies when you use them, not in a panic before an audit. That makes the record legitimate: it reflects real practice on real days.

It also keeps good practice moving between teachers. If you have a win with a challenging student and you log it, the next teacher of that student sees it immediately — and can ride the same strategy instead of starting from scratch or rediscovering what already worked.

Funding for special accommodations is substantial and audited regularly (often every other year). When evidence is captured as you teach, the chase for paper trails and teacher recollections largely disappears. Sentinel Administrators can then see compliance at a glance when audit preparation begins (see below).

> Good for teachers and students: clearer strategies, better relationships, authentic engagement — and wins that travel with the student across the timetable. Good for the school: less time lost to compliance admin — a huge time-saver that adds up across staff and years.

Sentinel Administrators

Sentinel Administrators are the leaders and office staff who oversee funded accommodations and audit readiness — Heads of Wellbeing, Deans, learning support coordinators, and similar roles. They use Sentinel Admin in the Suite (or the Sentinel app) while classroom teachers continue with day-to-day Sentinel on each class.

Monitor is the at-a-glance view: year-group heat maps show which funded accommodations are on track, under-delivered, or period-ended. Search by student or teacher, open a cell for detail, and see how many times each accommodation was logged in the period — evidence tied to the requirement, not a separate spreadsheet you have to reconcile.

That is the audit-preparation win. Instead of untold hours chasing teachers and stitching together folders, Administrators can see where the school stands before the auditors arrive. Records logged in Sentinel are already structured against plan lines and delivery evidence — ready to export and present in the compliant manner auditors expect.

- Monitor — year heat map, standing status (on track / under / period ended), and drill-down evidence
- Setup — add or edit funded accommodation plan lines when needed (TASS sync is the long-term source for volume)
- Search — find a student or teacher quickly during a compliance check
- Teacher logs in class — every authentic delivery in Sentinel feeds what Administrators see in Monitor

> Sentinel connects the classroom to compliance: teachers log in the moment for students; Administrators see the picture for the school. Less chasing, fewer gaps, faster audits — time saved is money saved.

What you record

On the class board you create and update student records from the toolbar. Each card has a brief (what matters) and strategies (what to do).

Action	Use it for
+ add info	Standing profile — Learning support, Accommodations, or Behaviour that stay relevant over time

Action	Use it for
+ concern	Concern — Academic, Behavioural, or Uniform — something staff need to notice and follow up
+ temp	Short-lived update — Pastoral, Accommodations, or Behaviour with an expiry (e.g. a bereavement this week)
+ log win	What worked — a strategy win other teachers of that student can see straight away and use in their next lesson

Log wins are especially powerful for challenging students: one successful approach in Period 2 can be on the next teacher's board by Period 3. You can later fold a proven win into the standing profile so it stays available over time.

You can paste a support plan or staff note and Summarise with AI into Brief / Strategies. Choose who can see each note (personal, other teachers, or school leaders such as Head of Wellbeing / Dean of Year / Head of Academia).

- Consult — peer-coach chat before or after class; save personal class notes or an accommodation audit log of what you delivered
- Day and class picker — work the tactical list for today's classes; temps fall off when they expire
- Import plan → Summarise — turn long documents into classroom-ready brief and strategies

How you use it day to day

1. Open Sentinel for the class you are about to teach (sidebar or S chip).
2. Scan standing profiles and any temps or concerns for today.
3. Teach with the strategies on the cards; Consult if you need a quick peer coach.
4. Log wins, temps, or concerns as they happen — not later.