Absolutely—let’s build this **dual-track** so *any* adult can run the class from day 1:

1. a **Student Lesson Schema** your app renders,
2. a matching **Teacher Guide Schema** (run-of-show, scripts, policies, grading),
3. **Console presets** (AI-tutor scope, XP, rubrics, materials) per lesson.

Below are production-ready schemas + fully filled examples for Lessons 1–3, plus configured presets for Lessons 4–18.

**1) Student Lesson JSON (app-facing)**

{

"version": "1.0",

"lesson\_id": "S1-L01",

"title": "Welcome to CodeQuest",

"week": 1,

"duration\_min": 60,

"cpalms": ["SC.912.CS-CS.1.1"],

"objectives": [

"Explain what algorithms are and identify real-life examples.",

"Build a first interactive program (confidence win)."

],

"materials": ["Scratch account", "Headphones (optional)"],

"activities": [

{

"type": "warmup",

"title": "Broken Robot",

"minutes": 10,

"student\_steps": [

"In pairs, give the 'robot' (teacher or peer) exact step-by-step directions.",

"Debug when instructions fail."

]

},

{

"type": "mini\_lesson",

"title": "Algorithms Everywhere",

"minutes": 10,

"content\_ref": "slides://algorithms-everywhere"

},

{

"type": "build",

"title": "About Me Interactive Poster (Scratch)",

"minutes": 25,

"student\_steps": [

"Create a new Scratch project.",

"Add 2 sprites, 1 sound, and 1 keypress event.",

"Animate your name or interests."

],

"success\_criteria": [

"At least 1 event (when key pressed/when sprite clicked).",

"At least 1 motion or looks block.",

"Project runs without errors."

]

},

{

"type": "share",

"title": "Spotlight Demos",

"minutes": 10,

"student\_steps": ["Volunteer to demo your project.", "Give one 'glow' and one 'grow' peer comment."]

},

{

"type": "close",

"title": "Exit Ticket",

"minutes": 5,

"prompt": "Name one algorithm you used today outside of class."

}

],

"homework": {

"title": "Tech Interview",

"instructions": "Interview a family member about daily tech use; list 3 apps and why they use them.",

"xp\_value": 15,

"due\_at": "P1-W02-23:59"

},

"assessment": {

"type": "artifact",

"rubric\_ref": "rubric://starter-project",

"submission": {

"kind": "link",

"instructions": "Share your Scratch link in the app."

}

},

"xp": { "project": 50, "homework": 15 },

"badges": [{ "when": "project\_submitted", "award": "FIRST\_BUILDER" }],

"tutor\_policy\_preset": {

"mode": "LEARN",

"scope\_level": "TIGHT",

"allowed\_topics": [

{ "cpalms": "SC.912.CS-CS.1.1", "slug": "algorithms-basics" },

{ "cpalms": "SC.912.CS-CP.1.2", "slug": "events-and-sequences" }

],

"tool\_whitelist": ["HINTS", "SNIPPETS", "SCRATCH\_HELPER"],

"blocklist": ["full solution request", "write my whole code"],

"snippet\_line\_cap": 0

}

}

**2) Teacher Guide JSON (run-of-show + controls)**

{

"version": "1.0",

"lesson\_id": "S1-L01",

"title": "Welcome to CodeQuest",

"week": 1,

"run\_of\_show": [

{ "t": 0, "min": 2, "script": "Welcome students. Today you’ll build your first interactive program in Scratch." },

{ "t": 2, "min": 8, "script": "Warmup 'Broken Robot': Take 2 volunteers. Enforce step-by-step instructions." },

{ "t": 10, "min": 10, "script": "Mini-lesson: show slides://algorithms-everywhere. Ask: Where do you see algorithms (TikTok, Maps)?" },

{ "t": 20, "min": 25, "script": "Build time: circulate; approve minimum spec (event + animation + sound)." },

{ "t": 45, "min": 10, "script": "Demos: call 3–4 volunteers; model 'glow/grow' feedback." },

{ "t": 55, "min": 5, "script": "Exit Ticket in app. Announce Homework. Remind submission steps." }

],

"console\_presets": {

"tutor\_mode": "LEARN",

"scope\_level": "TIGHT",

"rate\_limit\_per\_min": 3,

"max\_tokens": 700,

"allowed\_topics": ["algorithms-basics", "events-and-sequences"],

"blocklist\_patterns": ["write the entire", "do my assignment"],

"assessment\_lock": false

},

"rubric": "rubric://starter-project",

"rubrics\_catalog": {

"rubric://starter-project": {

"criteria": [

{ "name": "Functionality", "weight": 0.4, "desc": "Meets min spec; runs error-free" },

{ "name": "Clarity", "weight": 0.25, "desc": "Readable naming/organization" },

{ "name": "Documentation", "weight": 0.15, "desc": "Short project note or title cards" },

{ "name": "Creativity", "weight": 0.2, "desc": "Visual flourish or unique idea" }

]

}

},

"iep\_accommodations\_tips": [

"Offer paired programming.",

"Allow extra time; reduce required sprites to 1."

],

"ell\_support": ["Provide visual vocabulary of 'event', 'sequence', 'sprite'."],

"checks": [

"Before class: test Scratch logins.",

"During build: approve min spec per student.",

"After class: post recap; auto-open HW."

],

"grading\_ops": {

"base\_xp": 50,

"late\_policy": "minus10\_per\_day\_floor50",

"early\_bonus": "plus5\_if\_24h\_early",

"creativity\_bonus\_pct\_cap": 10

},

"evidence\_pack": {

"cpalms": ["SC.912.CS-CS.1.1"],

"artifacts": ["Scratch links", "Exit ticket data"],

"export\_on": "weekly"

}

}

**Lessons 2–3 (both tracks) — fully filled**

**Lesson 2 — Loops & Events (Clicker Game)**

**Student JSON (condensed):**

{

"lesson\_id": "S1-L02",

"title": "Loops & Events: Clicker Game",

"week": 2,

"duration\_min": 60,

"cpalms": ["SC.912.CS-CP.1.2"],

"objectives": [

"Use loops to repeat actions.",

"Use events to trigger behavior."

],

"activities": [

{ "type": "warmup", "title": "Loop Clap", "minutes": 5 },

{ "type": "mini\_lesson", "title": "Loops Visualized", "minutes": 10, "content\_ref": "slides://loops-scratch" },

{

"type": "build",

"title": "Clicker Game",

"minutes": 35,

"success\_criteria": ["Score variable increments", "At least 1 event", "No runtime errors"]

},

{ "type": "close", "title": "Reflection Card", "minutes": 10, "prompt": "What did your loop do for you?" }

],

"homework": { "title": "Power-Up Mechanic", "xp\_value": 15, "due\_at": "P1-W03-23:59" },

"assessment": { "type": "artifact", "rubric\_ref": "rubric://starter-project" },

"xp": { "project": 50, "homework": 15 },

"badges": [{ "when": "project\_variation\_powerup", "award": "BUG\_SLAYER" }],

"tutor\_policy\_preset": {

"mode": "LEARN",

"scope\_level": "TIGHT",

"allowed\_topics": [

{ "cpalms": "SC.912.CS-CP.1.2", "slug": "loops" },

{ "cpalms": "SC.912.CS-CP.1.2", "slug": "events" }

],

"tool\_whitelist": ["HINTS", "SCRATCH\_HELPER"],

"blocklist": ["entire game code", "paste full solution"],

"snippet\_line\_cap": 0

}

}

**Teacher JSON (condensed):**

{

"lesson\_id": "S1-L02",

"run\_of\_show": [

{ "t": 0, "min": 5, "script": "Loop Clap energizer." },

{ "t": 5, "min": 10, "script": "Mini-lesson loops/events; demo a simple loop." },

{ "t": 15, "min": 35, "script": "Build: Clicker Game; approve score variable + event + loop." },

{ "t": 50, "min": 10, "script": "Reflection + preview leaderboard (Top 5 only next week)." }

],

"console\_presets": {

"tutor\_mode": "LEARN",

"scope\_level": "TIGHT",

"allowed\_topics": ["loops", "events"],

"assessment\_lock": false

},

"rubric": "rubric://starter-project",

"checks": ["Verify event + score variable.", "Record common issues for tutor tips."]

}

**Lesson 3 — Intro Python: Number Guessing**

**Student JSON (condensed):**

{

"lesson\_id": "S1-L03",

"title": "Intro to Python: Number Guess",

"week": 3,

"duration\_min": 60,

"cpalms": ["SC.912.CS-CP.1.3"],

"objectives": [

"Transition from blocks to text coding.",

"Use variables and input/output in Python."

],

"materials": ["Replit (Python)", "Starter snippet"],

"activities": [

{ "type": "warmup", "title": "Blocks → Code Mapping", "minutes": 10 },

{ "type": "mini\_lesson", "title": "Variables & input()", "minutes": 10, "content\_ref": "slides://py-variables" },

{

"type": "build",

"title": "Number Guess Game",

"minutes": 30,

"success\_criteria": [

"Stores a secret number",

"Compares user input",

"Prints win/try again"

]

},

{ "type": "close", "title": "Pair Test & Reflect", "minutes": 10 }

],

"homework": { "title": "Add Hints (too high/low)", "xp\_value": 15, "due\_at": "P1-W04-23:59" },

"assessment": { "type": "artifact", "rubric\_ref": "rubric://python-basics" },

"xp": { "project": 50, "homework": 15 },

"badges": [{ "when": "project\_submitted", "award": "PYTHON\_PRODIGY" }],

"tutor\_policy\_preset": {

"mode": "LEARN",

"scope\_level": "TIGHT",

"allowed\_topics": [

{ "cpalms": "SC.912.CS-CP.1.3", "slug": "variables" },

{ "cpalms": "SC.912.CS-CS.2.5", "slug": "control-flow-basics" }

],

"tool\_whitelist": ["PY\_SNIPPETS", "HINTS"],

"blocklist": ["entire program", "solve full assignment"],

"snippet\_line\_cap": 12

}

}

**Teacher JSON (condensed):**

{

"lesson\_id": "S1-L03",

"run\_of\_show": [

{ "t": 0, "min": 10, "script": "Show Scratch→Python mapping; normalize anxiety ('text code' is just precise blocks)." },

{ "t": 10, "min": 10, "script": "Teach variables, input(), int(), print()." },

{ "t": 20, "min": 30, "script": "Build time: Number Guess; circulate; require min spec." },

{ "t": 50, "min": 10, "script": "Pair test, swap laptops; exit reflections in app." }

],

"console\_presets": {

"tutor\_mode": "LEARN",

"scope\_level": "TIGHT",

"allowed\_topics": ["variables", "control-flow-basics"],

"snippet\_line\_cap": 12

},

"rubrics\_catalog": {

"rubric://python-basics": {

"criteria": [

{ "name": "Functionality", "weight": 0.4 },

{ "name": "Code Quality", "weight": 0.25 },

{ "name": "Documentation", "weight": 0.15 },

{ "name": "Creativity/UX", "weight": 0.2 }

]

}

},

"teacher\_checks": ["Replit accounts working", "Starter snippet distributed"],

"accommodations\_tips": ["Allow pair-programming; offer typed template."]

}

**3) Lessons 4–18 (configured presets, ready to seed)**

You can store these as rows; each has Student + Teacher presets, CPALMS codes, tutor scope, rubric, and XP.

* **L04 – Conditionals: Magic 8-Ball (Python)**  
  CPALMS: SC.912.CS-CS.2.5, SC.912.CS-CP.1.3  
  Tutor: LEARN/TIGHT, topics: conditionals, random.  
  XP: 50 project, 15 HW. Quiz #1 (20 XP).
* **L05 – Loops: Rock–Paper–Scissors (Python)**  
  CPALMS: SC.912.CS-CP.1.2  
  Tutor: LEARN/TIGHT, topics: loops, comparisons.  
  XP: 50 project, 15 HW.
* **L06 – Mini-Project + Mini-Test #1 (Python)**  
  CPALMS: SC.912.CS-CS.3.2  
  Tutor: ASSESS/TIGHT during test; LEARN otherwise.  
  XP: 100 project, 50 test.
* **L07 – Draft Day #1 (Teams) + Icebreakers**  
  CPALMS: SC.912.CS-CS.5.1  
  Tutor: LEARN/NORMAL (team norms, planning).  
  XP: 20 team bonus.
* **L08 – AI Basics: Teachable Machine**  
  CPALMS: SC.912.CS-CS.4.2  
  Tutor: LEARN/TIGHT (datasets, bias, model basics).  
  XP: 50 project. Quiz #2 (20 XP).
* **L09 – Team AI Mini-Project**  
  CPALMS: SC.912.CS-CS.5.1, SC.912.CS-CS.2.6  
  Tutor: LEARN/NORMAL (integration help; no full builds).  
  XP: 100 team (60/40 split).
* **L10 – HTML Foundations**  
  CPALMS: SC.912.CS-CS.2.6  
  Tutor: LEARN/TIGHT (HTML structure).  
  XP: 40 project, 15 HW.
* **L11 – CSS Styling**  
  CPALMS: SC.912.CS-CS.2.6  
  Tutor: LEARN/TIGHT (selectors, classes).  
  XP: 40 project, 15 HW.
* **L12 – AI Meme Page (Web)**  
  CPALMS: SC.912.CS-CS.2.6, SC.912.CS-CS.4.2  
  Tutor: LEARN/NORMAL (fetch text from AI safely; policy: no offensive content).  
  XP: 50 project.
* **L13 – Mini-Test #2 (Web)**  
  CPALMS: SC.912.CS-CS.2.6  
  Tutor: ASSESS/TIGHT (concept only).  
  XP: 50 test.
* **L14 – Draft Day #2 + Problem Selection**  
  CPALMS: SC.912.CS-CS.5.1  
  Tutor: LEARN/NORMAL (scoping product).  
  XP: 10 planning artifact.
* **L15 – Sprint 1 (Capstone)**  
  CPALMS: SC.912.CS-CS.3.2, .5.1  
  Tutor: LEARN/NORMAL (snippets ≤12 lines).  
  XP: 0–20 teacher creativity bonus possible.
* **L16 – Sprint 2 + Peer Review**  
  CPALMS: SC.912.CS-CS.5.1  
  Tutor: LEARN/NORMAL.  
  XP: Peer reviews (5 XP each, cap 15/wk).
* **L17 – Playtest & Polish**  
  CPALMS: SC.912.CS-CS.3.2  
  Tutor: LEARN/TIGHT, focus: bug-fixing, UX copy.  
  XP: 0–20 teacher bonus.
* **L18 – Final Showcase (Capstone)**  
  CPALMS: Multiple (artifact + collaboration).  
  Tutor: ASSESS/TIGHT during grading windows.  
  XP: 200 team (split 60/40). Badges: FINAL\_BOSS, TEAM\_MVP.

**4) Teacher Console: weekly checklist (built-in)**

* **Before class:**
  + One-click “Load Lesson Preset” → sets tutor mode/scope, allowed topics, rubric, XP.
  + Materials check (accounts, slides, starter code).
  + Accessibility auto-notes (IEP/ELL hints).
* **During class:**
  + Run-of-show timer, activity cards, “Approve Min Spec” checklist.
  + Live tutor transcript + moderation pane (approve/deny queued requests).
  + Quick XP bonus (+5/+10/+20) with audit reason.
* **After class:**
  + Auto-open homework; post recap.
  + Grade queue (rubric quick sliders) → XP events emitted.
  + Standards Evidence Pack export (CPALMS + artifacts).
  + Leaderboard snapshot scheduled.

**5) Rubrics Catalog (IDs you can seed)**

* rubric://starter-project (Functionality 40, Clarity 25, Docs 15, Creativity 20)
* rubric://python-basics (Functionality 40, Code Quality 25, Docs 15, Creativity 20)
* rubric://web-basics (Structure 30, Styling 30, Accessibility 20, Creativity 20)
* rubric://team-capstone (Functionality 30, Impact 20, Code/Design Quality 20, Collaboration 20, Presentation 10)

**6) Ready-to-import seed lists**

* **Badges:** FIRST\_BUILDER, BUG\_SLAYER, PYTHON\_PRODIGY, AI\_TINKERER, PIXEL\_PIONEER, TEAM\_MVP, STREAK\_TOP5\_3, STREAK\_TOP5\_6, FINAL\_BOSS.
* **Tutor tool codes:** HINTS, SNIPPETS, SCRATCH\_HELPER, PY\_SNIPPETS, HTML\_HELPER, AI\_CONCEPTS\_ONLY.
* **Blocklist starter patterns:** “write the entire”, “paste full solution”, assignment titles.