

Build Your Own Economics Games and Simulations with AI

A step-by-step guide for high school and college teachers

Florencia Gabriele, PhD · Berklee College of Music · Lovable Ambassador



START HERE · FREE LOVABLE CREDITS

Sign up with this link to get extra credits and try every prompt in this guide.

lovable.dev/invite/R27A27R

This guide takes you from an idea to a classroom game you can project or share with a QR code. Sections 1 to 3 take about 30 minutes and cost no credits. Section 4 is the build. Sections 5 and 6 are copy-and-paste prompts and ideas for economics.

1. What is vibe coding?

Vibe coding means building software by describing what you want in plain language. The AI writes the code; you test it and steer it with more plain-language requests. Andrej Karpathy, a founding member of OpenAI and former Director of AI at Tesla, coined the term in February 2025, and Collins Dictionary named it Word of the Year for 2025.

You do not need to know how to code. You need what good teachers already have: a clear goal, a task broken into steps, a sense of where students get stuck, and the habit of giving specific feedback.

Which tool for what

Type	Examples	Best for	You get
Chat canvases	Claude artifacts, Gemini Canvas, ChatGPT code blocks	One-screen tools: a timer, a quiz, an interactive graph	A single web page
App builders	Lovable, Replit, Bolt	Real apps: saved data, logins, AI inside, a stable link	A published web app
Coding agents	Claude Code, Cursor	Large projects, for people comfortable with code	A codebase to maintain

This guide uses Lovable because a live class experiment needs a shared database and a link every student can open. For a one-screen tool, a chat canvas may be enough.

2. Set up your Lovable account (10 minutes)

1. Open <https://lovable.dev/invite/R27A27R> (or scan the QR code above) and sign up with your email or Google account.
2. The free plan gives you 5 credits a day, up to 30 a month. Daily credits do not roll over.
3. Teachers and students with a school email can get 50% off Lovable Pro for up to 12 months: lovable.dev/students.
4. Students under 18: Lovable's terms require users to be 18 or to have a parent's or guardian's permission. For grades 6–12, imagi Edu provides free, anonymous classroom accounts with no student data shared: lovable.dev/classroom.
5. How credits are spent: Lovable has three modes (Chat, Plan and Build). Each Plan mode message costs 1 credit plus any research; Chat mode costs less; in Build mode the cost depends on the size of the change. Thinking first saves credits.

3. Design before you build (15 minutes, no credits)

The best classroom games start from a misconception, not from a cool app. Answer these five questions on paper or in a chatbot before you open Lovable.

Question	Example: price ceilings
1. What concept do my students get wrong?	They think a price ceiling helps everyone who wants the good.
2. What will students decide?	Buyers and sellers set prices; the teacher moves a legal maximum price.
3. What happens when they get it wrong?	They see the shortage and the trades that never happen, with a short explanation.
4. How does it end?	A summary screen and three debrief questions.
5. Where is it played?	On the projector for the class, and on phones for the market.

Choose the level of play

Level	Type	Example	Needs
1	Review game	Shortage or Surplus?: teams read a market scenario and call it before the timer runs out	Screen only
2	Simulation	Move a price ceiling and watch the shortage and deadweight loss appear	Screen only
3	Live experiment	The whole class trades on phones; the price converges on the projector	Database (Lovable Cloud)
4	AI tutor	Asks why the shortage appeared and never hands over the answer	AI and database

Start with level 1 or 2. Move up once you are comfortable with the build loop.

4. Build it, step by step

Lovable updates its menus often. If a button looks different, trust your screen; the steps stay the same.

Step 1. Create a project and add your design rules

Lovable's Project knowledge is sent with every prompt, so you write your rules once and every game in the project follows them. Open your project's Project knowledge section and paste the rules below.

They put motivation research to work: points that only go up, teams instead of rankings, instant feedback, cheap mistakes, and choices.

PROJECT KNOWLEDGE · COPY AND PASTE

Design rules for every game in this project:

- Points start at 0 and only go up.
- Teams compete; no individual ranking is shown.
- After every answer, give instant feedback: what was right or wrong, and why.
- Mistakes are cheap: allow a retry with a hint.
- Offer two paths (core and challenge) to the same learning goal.
- The game mechanic is the economic concept itself, never a decorative reward.
- Frame it as a short story: the student plays a role (producer, consumer, central banker).
- Large type and high contrast; it must work on a phone and on a projector.
- Never ask for names or personal data.

Step 2. Write the first prompt as a story

Tell Lovable who uses the screen, what they decide, what they should learn and how it should feel.

Paste your real content (terms, questions, numbers) instead of placeholders. Ask for one screen, with no login and no database, unless you need them.

TEMPLATE

```
I teach [subject] to [level and age]. My students need to practice [one concrete goal].  
Build one screen: a [game type] where the student [key action].  
Use this content: [your real list].  
When they get it right, [feedback]. When they get it wrong, show a short explanation  
and let them try again.  
At the end, show the score and what to review.  
Tone: [playful, calm, serious]. No login, no database, no personal data.
```

EXAMPLE · SHORTAGE OR SURPLUS?

```
I teach economics to high school and college students. Build a one-screen classroom  
game called Shortage or Surplus?, made to be projected.  
I paste a list of market scenarios, one per line, each followed by its answer:  
shortage, surplus or no change. Show one scenario at a time in large text, with a 45-  
second timer. Teams of 4 to 5 take turns; the team on turn picks one of three big  
buttons: Shortage, Surplus or No change. Then reveal the answer with a one-sentence  
explanation. Points start at 0 and only go up; show each team's score. At the end, list  
the scenarios teams missed so we can review them together.  
Use these scenarios to start: A city caps rents below the market rent → shortage. A  
minimum wage is set above the equilibrium wage → surplus (of labor). A price ceiling is  
set above the equilibrium price → no change.  
High contrast, readable from the back of the room. No login, no database, no personal  
data.
```

Step 3. Use Plan mode for anything big

Plan mode thinks with you before writing code: it asks questions and proposes a plan you can edit. Use it at the start of a project, before a big feature, and whenever something breaks. It costs 1 credit per message, plus any research.

LET LOVABLE INTERVIEW YOU

```
I want to build [idea] for my [level] economics students. Before you build anything,  
ask me the five questions you need answered to build this well. Don't write code yet.
```

Step 4. Build in small loops

- **One change per prompt.** Check the preview, then send the next one. If your prompt has more than two “and”s, split it.
- **Say what not to touch.** End prompts with a guardrail such as “Don’t change anything else.”

- **Point instead of describing.** For colors, sizes and text, select the element in the preview and change it directly.
- **Bookmark what works.** Every change is saved in version history. Bookmark the version after each working feature so you can always go back.

EXAMPLE OF A SMALL LOOP

Add a screen before the game where I can type the team names. Don't change the game screen or the scoring.

Step 5. When something breaks

1. Click “Try to fix” once or twice.
2. If the error is still there, stop retrying. Switch to Plan mode and ask for the root cause (prompt below).
3. If a change made things worse, restore the last bookmarked version from the history, or edit your earlier message and choose “Revert and resend.”
4. If the project is tangled, remix it: you get a clean copy and the original stays safe.

FIND THE CAUSE FIRST

Investigate why [the timer doesn't reset between rounds]. Find the root cause before changing any code, and explain what you find.

Step 6. Test it like a student

- Switch the preview to phone size and play the whole game.
- Project it: can the back row read it?
- Try wrong answers, empty fields and double clicks.
- If it saves data, refresh the page and check that the data is still there; open it on two devices at once.
- Check every number, curve and “correct” answer yourself. A polished app can still compute the wrong equilibrium.

Step 7. Publish and share

1. If the app stores anything, run Lovable’s security scan and fix anything marked critical before you share the link: the scan does not block publishing.
2. Click Publish (top right). Your app goes live at a lovable.app address.
3. After later changes, click Publish, then Publish changes. Students only see the version you publish.
4. Share the link, or show a QR code. You can ask Lovable to add a QR code with the link to the start screen.

Share link vs. publish. A share link shows your work in progress to collaborators and expires. Publishing is what puts the app online for students.

Step 8. Play, debrief and measure

The game is not the lesson; the debrief is. Use the same rhythm every time: **predict, play, explain.**

- Before playing, ask for a prediction: “What will happen to the price if...?”
- After playing: What did you predict, and what happened? Why did it happen? Who gained, who lost, and what was left on the table?
- Around the world in 80 seconds: go through the key terms; each student adds one fact until the concept is exhausted.
- To know whether it worked, ask Lovable to add one anonymous question before and after the game, and compare. Check who benefited, not only the average.

5. Prompt library for economics

Copy, paste and adapt. Each prompt assumes the design rules from Step 1 are in your Project knowledge.

SIMULATION · PRICE CEILING (LEVEL 2)

Build a one-screen price ceiling simulation for high school and college economics. Demand: $Q_d = 120 - 2P$. Supply: $Q_s = 3P - 30$. Equilibrium: $P = 30$, $Q = 60$. A slider sets a price ceiling from 10 to 40. When the ceiling is below 30, show quantity supplied, quantity demanded and the shortage, and shade consumer surplus, producer surplus and deadweight loss on the graph, with the numbers updating live. Before the reveal, students type a prediction. A College mode toggle shows the equations and the elasticities at equilibrium. No login.

LIVE CLASSROOM MARKET (LEVEL 3, NEEDS LOVABLE CLOUD)

Build a live classroom market for one concert ticket. Two views. Teacher view, made for a projector: a large QR code and join link, live counts of buyers and sellers, and a Reveal button that draws the demand curve (buyers' maximum prices, highest to lowest) and the supply curve (sellers' minimum prices, lowest to highest), marking the price and quantity where they cross. Student view on a phone: no login; on joining, randomly assign Buyer or Seller; buyers enter the most they would pay and sellers the least they would accept, from 0 to 200 dollars; one answer per device. Store only the role and the price, never names. A Reset button starts a new round. Large type, high contrast.

AI TUTOR (LEVEL 4, ADD TO THE SIMULATION)

Add a tutor panel using Lovable's built-in AI. The tutor asks the student why the shortage appeared, one question at a time, and never states the answer. It stays on supply, demand and price controls, uses plain English, never asks for personal information, and ends after 10 messages.

ONE GAME, ALL YEAR

Add a teacher screen where I can paste my questions as a table (question, options, correct answer, explanation), and make the game use them. Don't change the design or the game logic.

PROJECTOR MODE

Add a Projector mode button that shows the current question full screen, with very large text and high contrast, and no menus. Don't change the game logic.

POLISH PASS

Take a critical pass across the whole app. Tighten typography and spacing. Improve empty and loading states. Don't add new features. Don't change colors unless something is unreadable.

Check the math. Before class, verify every number the app shows against your own calculation, especially surplus areas and equilibrium values.

6. Ideas for your economics class

By moment of the class

Moment	What to build
Before class	A short interactive explainer and one prediction question (flipped classroom)
Opening	Shortage or Surplus? or a quick quiz on last class's terms: retrieval practice
Core	A simulation or a live market: students decide and the model responds
Closing	Debrief questions and an exit ticket: predict, play, explain
After class	An AI tutor that only asks questions; calculators where students try their own scenarios without submitting their numbers
Assessment	Decisions students make and justify inside a scenario, instead of recall an AI can answer for them

By topic

Area	Ideas
Micro	Tax incidence; externalities and tradable permits; public goods; tragedy of the commons; monopoly and Cournot duopoly; prisoner's dilemma; auctions and the winner's curse; the market for lemons
Macro	The multiplier; money creation; a central bank game with an inflation target; aggregate supply and demand shocks; sorting GDP components and types of unemployment; saving under high inflation
Personal finance	Compound interest; credit card debt; budgeting; tax brackets; inflation and savings; diversification
AI inside the app	A Socratic tutor that never gives the answer; interview Adam Smith, Keynes or Hayek; feedback against your own rubric

7. Before you share the link

- **Check the economics:** test every answer, curve and number.
- **Protect students:** a published app is a public link. No names, grades or photos; use codes if you need a database.
- **Students under 18:** Lovable requires parent or guardian permission; for grades 6–12, use imagi Edu's free classroom accounts.
- **Security:** run the security scan if the app stores anything.
- **Devices:** test on a phone and on the projector.
- **Cost:** AI features inside your app are billed by use; check before a whole class uses them.
- **Backup:** keep a bookmarked version that works, and a plan B for the day the Wi-Fi fails.
- **Simplest tool:** if a single page is enough, a chat canvas may be faster than a full app.

8. Learn more

- **Lovable Academy** — learning paths, videos and templates: academy.lovable.app/academy
- **Ideas for educators** — one-click starting prompts: academy.lovable.app/academy/what-to-build
- **Five principles of prompting:** academy.lovable.app/academy/prompting
- **From idea to app** — the official step-by-step: docs.lovable.dev/tips-tricks/from-idea-to-app
- **Prompt library:** prompt-library-ui.lovable.app
- **Lovable for classrooms (imagi Edu):** lovable.dev/classroom
- **Hour of AI with Lovable and imagi**, December 7–11, 2026: imagilabs.com/hour-of-ai

- **Teacher Hive** — tools made by teachers: teacherhive.app
- **IA Today** — our AI newsletter for educators: newsletter.iatoday.ai

Free Lovable credits: <https://lovable.dev/invite/R27A27R> · Questions:
Florencia@florenciagabriele.com