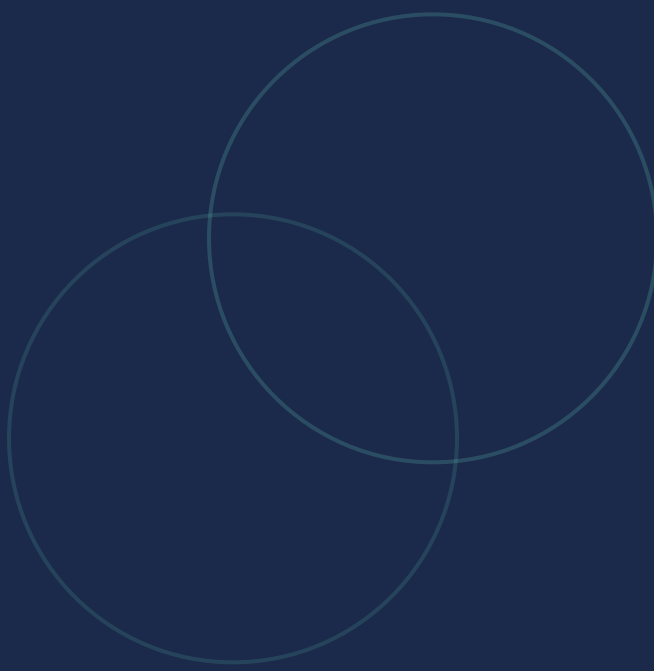


VIBE CODING 101

Build a real app, one prompt at a time



A 120-minute, hands-on training session where learners build a working single-page app in Lovable — one prompt, one action, one preservation phrase at a time.

What we're doing today

Five chapters across two hours. Most of today is hands-on, not lecture.



 **GOAL** By the end of today, you will have built a working single-page app in Lovable using six prompts, with preservation phrases protecting your work.

What is a single-page app?

Every SPA is made of three kinds of blocks. Once you can name the block, you can prompt for it.



Visual Blocks

Hero, navbar, cards, animations.

What the user sees.



Interaction Blocks

Buttons, forms, tabs, search.

How the user acts.



Data Blocks

State, localStorage, backend.

What the app remembers.

 **Analogy** Think of an SPA like a vending machine — visual is the glass front, interaction is the buttons, data is the inventory behind the panel.

The prompt formula

Action + Target + Constraint + Preservation Rule

ACTION

"Add a hero banner..."

TARGET

"...below the header"

CONSTRAINT

"Use one CTA button"

PRESERVATION

"Do not redesign the page"

✓ STRONG PROMPT EXAMPLE

"Add a three-card section below the hero. Each card explains one benefit: speed, creativity, iteration. Do not redesign the existing hero."

One Prompt, One Action.

Small prompts beat one giant prompt. Every time.

✗ One Giant Prompt

"Build me an app with a hero, carousel, login, dashboard, animations, a database, and dark mode in one shot."

Lovable guesses. Generates a sprawling mess. One follow-up wipes out what worked.

✓ One Prompt, One Action

"Add a hero banner with headline, subheadline, and one CTA button. Do not add other sections yet."

Lovable produces a clean, scoped output. Easy to review. Easy to build on.

 THE WORKFLOW LOOP

Prompt



Generate



Inspect



Test



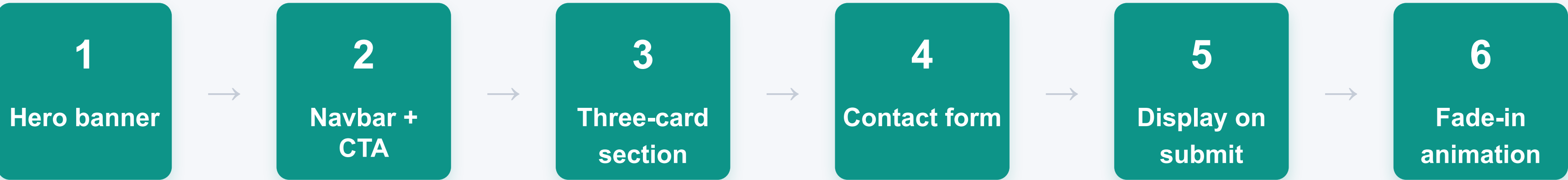
Improve



Commit

Plan, then prompt.

Small prompts only work if you know what you're building next.



 **THE PLAN** Your app today. Six features, six prompts. We'll build them in this order, one at a time. The plan is what makes small prompts possible.

Lesson 1 · Hero Banner

Live demo, then your turn. Same prompt structure, your project.

 10 min demo · 17 min practice

1

Type into Lovable:

"Create a single-page app for a student coding club."

2

Add the constraint:

"Add a hero banner with a headline, subheadline, and one call-to-action button."

3

Lock the scope:

"Do not add any other sections yet."

 **TRAINER NOTE** Type slowly. Hit send. Stay **silent** while Lovable generates — let learners watch the workflow loop happen.

Quick check.

Temperature read before we add the second prompt. Don't move on if more than a third are thumbs sideways.



Thumbs up

Your hero appeared with headline, subheadline, and one CTA.



Thumbs sideways

Lovable added extras you didn't ask for.



Hand up

You're stuck on the prompt itself. We'll come to you.

SHARE-OUT

Read your CTA button text out loud. Anyone want to share?

Lesson 2 · Navbar + Preservation

Two prompts back to back. Watch what the second sentence of Prompt B does.

 10 min demo · 18 min practice

A The navbar

"Add a navigation bar with a logo on the left and three links on the right: Home, Projects, Contact."

B The second CTA, preserved

"Add a second button in the hero section called 'View Projects.' Keep the existing button unchanged."

THE PRESERVATION PHRASE DOING THE WORK

"Keep the existing button unchanged" — that single sentence is the difference between an app you can keep building and an app that gets rewritten on every prompt.

Lesson 3 · Three-Card Section

Speed, creativity, iteration — one card each. Hero and navbar stay untouched.

 8 minutes

1

Type into Lovable:

```
"Add a three-card section below the hero. Each card should explain one benefit of learning vibe coding: speed, creativity, and iteration."
```

2

Look for:

Three cards in a row, hero and navbar untouched.

3

If anything above the cards shifts, add and rerun:

```
"Do not redesign the page."
```



DISCUSSION

What benefit would you put on a fourth card?

Lesson 4 · Contact Form

Front-end only. We wire data in 201, not today.

 8 minutes

1

Type into Lovable:

```
"Add a simple contact form with name, email, and message fields. Do not connect it to a database yet."
```

2

Look for:

A form with three fields plus a submit button. No backend wiring.

3

Notice the constraint:

"Do not connect it to a database yet" keeps it simple while you learn the pattern.



DISCUSSION

Why might "not yet" be a useful phrase to keep in your toolkit?

Lesson 5 · Display on Submit

Front-end state only. Then refresh — and watch it disappear.

 7 minutes

1

Type into Lovable:

"When the form is submitted, display the submitted message below the form using front-end state only. Do not use a backend or database."

2

Test it:

Type into the form. Click submit. Message appears below.

3

Then refresh the page:

It's gone. That's front-end state — no permanent save.



DISCUSSION

What's the difference between "state" and "database" in your own words?

Lesson 6 · Fade-in Animation

This is where preservation phrases earn their keep.

 8 minutes

1

Predict first:

What might Lovable change if you dropped the second sentence?

2

Then type:

```
"Add a subtle fade-in animation to the feature cards only. Do not change layout, text, or form behavior."
```

3

Watch:

Cards fade in on load. Everything else stays untouched.



DISCUSSION

If your form changed anyway, what phrase would you add to your prompt?

Pair share: what did Lovable rewrite?

Two rotations. 2.5 minutes per side. Same three shares, fresh ears.

 10 minutes

1 YOUR BUILD

Show your app to your neighbor.

2 WHAT LOVABLE REWROTE

Name one moment Lovable changed something you didn't ask for.

3 PRESERVATION PHRASE

Name the phrase you'd add next time to prevent that rewrite.

ROTATION 1

2.5 min · A shares

→

2.5 min · B shares

· switch partners ·

ROTATION 2

Same shares, fresh pair

Three things to remember.



One Prompt, One Action

Small, scoped prompts beat one giant prompt. Every time.



The prompt formula

Action + Target + Constraint + Preservation Rule. All four parts, every time.



Preservation phrases protect your work

"Do not redesign the page"

"Keep existing functionality unchanged"

"Make the smallest change possible"

OPEN Q & A

Questions?

Open Lovable tomorrow. Add one feature. **One prompt at a time.**

lovable.dev

replit.com

github.com