



Vibe Coding Lunch & Learn - Working Sheet



This worksheet was created by [Dan McKenzie](#) for a cross-team Lunch & Learn at [Risk Ledger](#) on AI & "vibe coding." It's designed for all abilities - whether you've never touched AI before or you've been building for years.

Goal: Ship a tiny app in an hour. Learn how to use AI to move from idea → working thing fast, without getting lost in process.

Last updated: @July 29, 2025

👁️ What is Vibe Coding (+ prompting)?

🚀 What we'll build & setup

The four rounds (ready-to-copy prompts)

- 1 Basic Win Logger
- 2 Users, Emoji, Timestamps
- 3 Styling, Total Badge, Persistence
- 4 Power-Up / Personalise (your pic)

🏆 Reflections & Present Back

📖 Further Reading

👁️ What is Vibe Coding (+ prompting)?

- Anyone can talk about ideas. Shipping is what counts.

- Speed in product isn't about cutting corners - it's about finding out what works before you've burned months on theory, process, or consensus. I once shipped a prototype 20 minutes before a meeting, straight off a Slack DM. No spec, no backlog. Built it, demo'd it live, and knew instaLIntly if it landed or flopped. That's how you find truth fast. That's how you build momentum.
- Vibe coding is using AI to go from idea → working thing fast, then refine. You don't need a spec or a committee - you need something people can click, test, and talk about. It's about:
 - Iterating, not theorising
 - Focusing on the next move, not the perfect answer
 - Thinking in systems without freezing up
- Prototyping is easy. Shipping something robust, safe, and customer-ready still takes time and a team. This is about getting to "is this worth it?" before you burn six months working the wrong thing.
- Say your intent, add constraints, describe the output shape, give the smallest useful context, and ask for one next step. Iterate like a pair: ask "what changed and why?", critique, adjust. Keep loops small and tight - smaller prompts beat sprawling ones for speed and quality. Be deliberate about speed vs depth: slow reasoning costs more and takes longer, so use it on purpose. Keep a scratchpad of prompts, share runs, and never merge AI-generated code without review.



What we'll build & setup

- We're making a tiny web app called **Win Logger** - a simple place where people log small wins in real time. It's deliberately lightweight so we can focus on *how* we build, not wrestling with scope.
- We'll take it through **four fast iterations**: start bare-bones, add features, improve UX & end with something you'd actually share. Each round is about making one meaningful change, shipping it, and seeing if it moves us forward.
- We'll be using (pick one):

- **Loveable** → <http://loveable.dev/>
- **Figma Make** → <http://figma.com/make>
- **Bolt** → <http://bolt.net/>
- At the end, you'll have gone from zero → working app in under an hour, learned the mechanics of vibe coding and seen how AI + no-code/low-code tools can get you to "something real" faster than you think.

The four rounds (ready-to-copy prompts)

Ship one meaningful change each round. After you've done it: post a link/screenshot + one sentence - what changed and why.

1 Basic Win Logger

Goal: Add a win → it appears instantly at the top of the list.

Create a simple web page **with:**

- A text input labelled "What did you achieve?"
- A submit button labelled "Add"
- A live-updating list showing all submitted wins, most recent at the top

Keep it minimal and clean. No extra features yet.

2 Users, Emoji, Timestamps

Goal: Each win shows name, emoji (if any), and timestamp.

Update the app to include:

- A field to capture the user's **name** (or initials)
- Optional emoji per **win** (e.g., pick from a small **set** or accept **:emoji:** syntax)
- Each win shows the time it was **added** (timestamp)

Maintain live updates. Keep the **UX** simple and fast.

3 Styling, Total Badge, Persistence

Goal: Refresh doesn't lose data; total badge updates live.

Update the app to include:

- A badge at the top showing the total number **of** wins submitted so far
- Dark mode **with** a bold, modern font and card-style containers **for** each win
- Persist wins **in** localStorage so the list survives page refresh

Keep interactions snappy and accessible.

4 Power-Up / Personalise (your pic)


Goal → Pick one upgrade (or invent your own) and push it further.

- Ideas:
 - Copy a win or share via email/Slack
 - Top users by total wins
 - Let users edit or remove their own wins

- Filter by user, emoji, or date
- Confetti or "👏" on add
- Export the win list as CSV or screenshot
- Make it responsive and lovely on mobile
- Show a positive message when a win is added
- Avg wins per user, most popular emoji, busiest hour
- Anything that would delight you



Reflections & Present Back

- You've just shipped four iterations in under an hour.
- Post link/screenshot in **#lunch-learn** 
- Answer quickly:
 1. What did you learn?
 2. What would you add next?
 3. Where else at work would this approach help?



Further Reading

- [How to Get Started with Vibe Coding with AI \(The Easy Way\).](#)
- [The Ultimate Vibe Coding Guide](#)
- [12 Rules to Vibe Code Without Frustration](#)
- [What is Vibe Coding? \(GeeksforGeeks\).](#)
- [A Beginner's Step-by-Step Guide to Vibe Coding](#)



Created by Dan McKenzie - see what we built in this LinkedIn post or find me at mckenz.io. I'd love to see what you build - share your creations with me.