BUILD PROJECTS · BUILDER · 20 MIN READ

Capstone: build and ship your first real app

A step-by-step project that uses everything in the library: plan it, build it, save it, store data safely, put it online and share it.

Pass the check (4 of 5) online to earn the “Shipped It” badge:
https://digitalbuilders757.com/university/study/capstone-build-and-ship

The project

Build a small, real tool someone you know will use in the next week. Good first projects: a sign-up list for a club or event, a booking request form for a local business, a shared reading or workout tracker, a price-quote calculator for a side hustle.

Finished means: it has a real web address, people can sign in, their data is saved and protected, and at least one real person used it.

Stage 1: Plan (30 minutes)

- Problem sentence: who struggles to do what, and why.
- The one job the app does, and three features you're cutting for now.
- The screens: list them (for example: sign in, list, add item, detail).
- The data: what tables you need and who may see each row.
- Ask an AI to challenge your plan: 'What's missing? What's too big for a first version?'

Stage 2: Build (2 to 4 hours)

- Start in a builder (Lovable, Bolt, v0 or Replit) with your plan as the first prompt; ask for sample data first.
- One feature per prompt; test each one like a user, on your phone too.
- Connect GitHub and commit every time something works.
- When something breaks, use the debugging steps: reproduce, find the error, brief the AI, fix small.

Stage 3: Real data, safely

- Connect Supabase (or your builder's database) and turn on sign-in.

- Turn on row level security for every table; add the smallest policies (users see their own rows, or the club admin sees everyone's).
- Test it: sign in as two different users and make sure neither can see the other's private data.
- Search your code for keys and secrets; only the public key may be in the front end.

Stage 4: Ship and share

- Publish from the builder or deploy the GitHub repo to Vercel; add your environment variables there.
- Optional: connect a domain you own.
- Write a short README: what it does, who it's for, how to run it.
- Give it to one real user. Watch them use it without helping, and write down where they got stuck.
- Ship one improvement based on what you saw. You're now a builder with a real project for your portfolio.

Key terms

MVP (minimum viable product)
The smallest version that solves the core problem well enough to use.

GitHub
A website that hosts Git repositories online for backup, sharing and teamwork.

Supabase
A back-end platform on Postgres with auth, instant APIs, storage, edge functions and realtime.

Deploy
Publish a version of your app to a host.

README
The front page of a repository: what it is and how to run it.

Prompt
What you give the AI: instructions, a question and any material to work from.

Commit
A saved snapshot of changes with a message.

Row level security (RLS)
Database rules that decide which rows each user may read or change.

Environment variable
A setting or secret the app reads at run time, kept outside the code.

Portfolio
A collection of real projects that shows what you can build.

Try it: Your capstone checklist

1. Plan written (problem, one job, screens, data).
2. Built in small steps, committed to GitHub.
3. Sign-in on, RLS on every table, tested with two users, no secrets in the front end.
4. Live at a real address, README written, one real user tried it, one improvement shipped.

KNOWLEDGE CHECK

Capstone: build and ship your first real app

5 questions. Circle one answer for each; 4 or more correct passes. Answers are at the back.

1. What makes a good first capstone project?

A. A clone of a giant social network, so it looks impressive on a portfolio
B. A small, real tool someone you know will use this week
C. Something you'll never show anyone
D. An app with 50 features

2. How do you check your data is protected before sharing the app?

A. Trust the builder's security badge, which checks this for you
B. Make the app pretty
C. Sign in as two users and check neither sees the other's data
D. Turn off sign-in so there's no private account data left to protect

3. When should you commit to GitHub during the build?

A. Only at the very end, once the whole app is finished
B. Never
C. Only when it's broken
D. Every time something works

4. Where do your production secrets go when you deploy to Vercel?

A. Environment variables in the Vercel project settings
B. In the README
C. In a comment in the code
D. In the page title

5. What's the best thing to do after your first real user tries the app?

A. Never change it again, since it already works for them
B. Fix the spot where they got stuck
C. Add ten new features at once so they have more reasons to come back
D. Delete it and start a bigger version

Official sites

Lovable: https://lovable.dev
Supabase: https://supabase.com
Vercel: https://vercel.com
GitHub: https://github.com

Last reviewed 2026-11-25. AI tools change fast; check each tool's own site for current features, plans and prices.

ANSWER KEY

Capstone: build and ship your first real app

1. B: A small, real tool someone you know will use this week
Small and real beats big and imaginary: you'll finish it and learn from a real user.

2. C: Sign in as two users and check neither sees the other's data
Testing with two accounts proves your row level security actually works.

3. D: Every time something works
Small, frequent commits are save points you can return to.

4. A: Environment variables in the Vercel project settings
Hosts store secrets as environment variables, separate from the code.

5. B: Fix the spot where they got stuck
Real feedback turned into a small release is the build loop that makes products great.