BUILD SAFELY · BUILDER · 12 MIN READ

Privacy basics for builders: collect less, protect more

Know what personal data your app collects, keep only what you need, lock it down, delete what you don't need, and plan for mistakes.

Pass the check (4 of 5) online to earn the “Privacy Keeper” badge:
https://digitalbuilders757.com/university/study/privacy-basics-for-builders

Every sign-up form is a responsibility

The moment your app collects a name, an email or a phone number, you're looking after other people's information. The US Federal Trade Commission's guide for businesses boils it down to five habits: take stock, scale down, lock it, pitch it and plan ahead. They work just as well for a weekend project as for a company.

Take stock and scale down

- List every piece of personal data your app collects, where it's stored (database, sheet, email tool, analytics) and who can see it.
- Ask of every field: do we need this to do the job? If not, don't collect it. A birthday, address or phone number 'just in case' is risk with no benefit.
- Keep it only as long as you need it, and write that down.
- Don't paste real users' data into AI tools to test or debug; use fake data instead.

Lock it

Turn on row level security so people can only read their own rows, keep keys in environment variables, give teammates the least access they need, and turn on multifactor sign-in for your database, hosting and email accounts. Only admins should see full lists, and exports should go somewhere safe, not a shared folder anyone can open.

Pitch it, plan ahead, and say what you do

Delete old data and test accounts you no longer need, and let users ask for their data to be deleted. Write a short plan for what you'll do if data leaks: who you tell, how you lock things down, and how you'll fix the cause. Publish a plain-language privacy policy that matches what your app really does. Privacy laws differ by place and by who your users are (children, for example), so check what applies to you; this guide isn't legal advice.

Key terms

Personal data
Information about an identifiable person, like names, emails or grades; follows privacy rules.

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.

Policy (RLS)
One row level security rule, like 'users can read their own notes'.

Privacy policy
A page telling people what data you collect, why, and what you do with it.

Least privilege
Give each person, app or AI only the access its job needs.

Two-factor authentication (2FA)
A second check at sign-in, like a code from an app, on top of your password.

Try it: A privacy check for your app

1. List every field your app collects and mark each one 'need' or 'nice to have'.
2. Remove one 'nice to have' field.
3. Sign out and check that one user can't see another user's data.
4. Write a three-line plan for a data leak and a one-paragraph privacy policy draft, then ask an AI what you've missed.

KNOWLEDGE CHECK

Privacy basics for builders: collect less, protect more

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

1. What does 'scale down' mean for your app's data?

A. Collect and keep only the personal data you really need
B. Make the database smaller by deleting old backups and log files
C. Store everything just in case
D. Use smaller fonts on the form

2. Your sign-up form asks for a home address you never use. What should you do?

A. Keep it, it might be useful
B. Remove the field
C. Make it required
D. Sell it to advertisers

3. Best way to debug a bug with user data using AI?

A. Paste the real customer table into the chat
B. Email the data to a friend to check
C. Recreate it with fake data that has the same shape
D. Post it in a public forum

4. Which setting stops one user reading another user's rows?

A. Dark mode for the admin dashboard
B. A bigger server that keeps each user's data in its own memory
C. A longer domain name
D. Row level security with owner-only policies

5. Why write a data-leak plan before anything goes wrong?

A. So you can act fast if it does
B. Because leaks never happen to apps that use a big cloud provider
C. It's only needed by big companies with their own security team
D. To make the privacy policy longer

Official sites

FTC: Protecting Personal Information, a guide for business: https://www.ftc.gov/business-guidance/resources/protecting-personal-information-guide-business-0
Supabase docs: Row Level Security: https://supabase.com/docs/guides/database/postgres/row-level-security
CISA: Turn on multifactor authentication: https://www.cisa.gov/secure-our-world/turn-mfa

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

ANSWER KEY

Privacy basics for builders: collect less, protect more

1. A: Collect and keep only the personal data you really need
Data you never collect can't leak. Fewer fields means less risk.

2. B: Remove the field
If you don't need it to do the job, don't collect it.

3. C: Recreate it with fake data that has the same shape
Fake data with the same structure shows the bug without exposing anyone.

4. D: Row level security with owner-only policies
Database rules decide who can read what, whatever the app's screens show.

5. A: So you can act fast if it does
A short plan made calmly beats guessing in a panic.