AUTHENTICATION · ~45 SEC READ

# The auth is always broken — you won't know until it's a lawsuit

> **Vibe-coded apps share one flaw: authentication that looks right and functions right, but whose security model is Swiss cheese. The developer never sees it, because they only ever tested with one account.**

One user logs in — works fine. Ten users — still fine. A thousand concurrent users, and suddenly session tokens are colliding, JWTs aren't expiring, and user A is staring at user B's dashboard. That's not a bug, it's a lawsuit. You can recover from a slow app. You cannot recover from a data breach — so auth gets fixed **before** anything else ships.

The good news: you don't need a security team. You need three checks you run every time.

## The three checks

- **Never roll your own auth.** No matter how smart the model is, it cannot build a secure authentication system from scratch. Plug in a battle-tested provider — **Clerk, Supabase Auth, or Auth0** — that millions of users hammer every day. Your custom solution is battle-tested by you and your demo account.
- **Run the logout test.** Log in, copy the URL, log out, then paste the URL back. If the page still loads, you don't have auth — you have a door with no lock. Expired sessions have to actually expire.
- **Row-level security on every query.** Every single database query filters by user ID. No exceptions. If user A can see user B's data by changing a number in the URL, you have a data breach waiting to happen — the line between an app and a legal liability.
- **No secrets in the model's hands.** Passwords hashed, never plaintext or `MD5`; login attempts rate-limited; password-reset links expire; admin access enforced server-side, never a client-side flag anyone can flip.

THE RULE

**Auth is not a feature. It's the foundation. Get it right before anything else, because one breach erases everything else you've built.**