

AI email-reply app for small businesses

Built with an AI app builder · hosted database · fictional sample based on real client work

THE VERDICT

The app works, and the core idea is solid. It is not ready for paying customers yet. Today, one business can read another business’s customer emails, and anyone who opens the browser’s developer tools can spend the AI budget. Both are fixable in days, not weeks. Fix the four critical items before launch and the rest in the first month.

REVIEW AT A GLANCE

| AREA                  | RATING     | HEADLINE                                       |
|-----------------------|------------|------------------------------------------------|
| Security              | NOT READY  | Admin page reachable without logging in        |
| Data isolation        | NOT READY  | Businesses can read each other’s inboxes       |
| Secrets & config      | NOT READY  | AI key in the browser; old keys in git history |
| Code quality          | NEEDS WORK | Core reply logic copied into seven places      |
| Maintainability       | NEEDS WORK | Settings hard-coded per environment            |
| Performance           | READY      | Fast enough for launch traffic                 |
| Reliability & backups | READY      | Daily database backups already on              |
| Launch readiness      | NEEDS WORK | Debug mode on; no limit on AI usage            |

ISSUES FOUND

|                                  |                            |                                        |
|----------------------------------|----------------------------|----------------------------------------|
| 4<br>critical: fix before launch | 3<br>high: fix in week one | 2<br>maintainability: fix in month one |
|----------------------------------|----------------------------|----------------------------------------|

**1. AI key shipped to the browser****CRITICAL**

**FOUND** The AI provider key sits in `app.js`, which every visitor downloads. Anyone can copy it and spend your credits.

**FIX** Move AI calls to a server route that reads the key from the environment. Revoke and replace the old key.

**2. One business can read another's inbox****CRITICAL**

**FOUND** `GET /api/emails?business=1042` returns that business's emails to any logged-in user. Nothing checks that 1042 is theirs.

**FIX** Derive the business from the signed-in user on the server, never from the URL. Add a test that tries to read another account and expects a refusal.

**3. Database readable with the public key****CRITICAL**

**FOUND** Row-level security is switched off on the `emails` and `businesses` tables, so the public key the app ships with can read every row.

**FIX** Turn row-level security on and add policies so each user can only reach rows belonging to their business.

**4. Admin page anyone can visit****CRITICAL**

**FOUND** `/admin` is hidden from the menu but returns the full dashboard to logged-out visitors.

**FIX** Check the user's role on the server for every admin route, not just in the menu.

**5. Old keys still in the git history****HIGH**

**FOUND** Two API keys deleted from the code months ago are still readable in earlier commits.

**FIX** Rotate both keys, then remove them from the history. Add a pre-commit check so new secrets are blocked.

**6. No limit on AI replies****HIGH**

**FOUND** `POST /api/reply` accepts unlimited requests. One script could run up thousands in AI costs overnight.

**FIX** Rate-limit AI calls per user and per business, and set a monthly spending cap with your AI provider.

7. Debug mode on in production

HIGH

**FOUND** Errors show full stack traces, file paths and database details to visitors.

**FIX** Turn debug mode off, show a friendly error page and send the details to an error tracker only you can see.

8. The same reply logic pasted in seven places

MAINTAINABILITY

**FOUND** The function that builds AI replies exists in seven near-identical copies. A fix in one leaves six still broken.

**FIX** Consolidate into one shared module with tests, and point every caller at it.

9. Settings hard-coded per environment

MAINTAINABILITY

**FOUND** Database URLs and feature switches are written into the code, with separate copies for testing and live.

**FIX** Keep configuration in the environment, not the code, so the same build runs safely in testing and production.

PRIORITY PLAN

| WHEN          | FIX                                                                       | EFFORT   |
|---------------|---------------------------------------------------------------------------|----------|
| Before launch | 1–4: key off the browser, data isolation, row-level security, admin roles | 2–3 days |
| Week one      | 5–7: rotate keys and clean history, AI rate limits, debug mode off        | 1–2 days |
| Month one     | 8–9: consolidate the reply logic, move config to the environment          | 2–3 days |

WHAT THIS SAMPLE LEAVES OUT

This sample shows part of one review. The full 360° Go-Live Review checks every area on page 1 in depth, and the fixes are made, not just listed: you launch with a secure app and a codebase the next developer can work in.