BEFORE YOU GO LIVE

SAMPLE REPORT

PRODUCTION-READINESS AUDIT

Ember Ledger

Prepared for **Sample Client (fictional)**

NOT READY

SCOPE	Next.js 14 app · Prisma + Postgres · Stripe · deployed on Fly.io
REVIEWED	github.com/sample-org/ember-ledger @ a3f91c2 (fictional)
DATE	2026-07-25
DOCTRINE	14 domains · 24 checks

The verdict

Readable in 30 seconds: where this build stands against the go-live doctrine.

3	9	5	7
CRITICAL FAILS	HIGH FAILS	NEEDS REVIEW	PASSING

> Ember Ledger is a well-built prototype that is not yet safe to launch. Two critical failures — an unscoped invoice query that lets any signed-in user read another tenant's invoices, and a live Stripe secret key committed to the repo — are ship-blockers on their own. Several reliability and infrastructure checks could not be confirmed from the code alone and need the artifacts named below. The happy path is solid; the failure paths and the multi-tenant boundary are where this build is exposed.

Launch blockers

Every critical and high-severity failure, ranked. Each must be resolved or consciously accepted before go-live.

CRITICAL

Invoice queries are not scoped to the caller's tenant

AUTH-03 · Authentication & access control · high confidence

- `app/api/invoices/[id]/route.ts:18 — const inv = await prisma.invoice.findUnique({ where: { id: params.id } })`
- `app/api/invoices/[id]/route.ts:22 — return NextResponse.json(inv) // no orgId / userId filter`

FIX

Scope every invoice read/write to the caller's org: findFirst({ where: { id, orgId: session.orgId } }). Add a Prisma middleware or Postgres RLS policy so no data path can forget the tenant filter.

CRITICAL

Live Stripe secret key committed to the repository

SEC-01 · Secrets & credentials · high confidence

- `.env.production:4 — STRIPE_SECRET_KEY=sk_live_51Q...redacted...9f (committed; tracked by git)`
- `.gitignore:1 — .env.local // .env.production is NOT ignored`

FIX

Rotate the key in the Stripe dashboard immediately, remove .env.production from git and scrub it from history (git filter-repo), add .env* to .gitignore, and load the key from Fly secrets at runtime.

CRITICAL

Search endpoint builds SQL by string concatenation

INP-03 · Input validation & web vulnerabilities · high confidence

› `lib/search.ts:11 — const rows = await db.$queryRawUnsafe(`SELECT * FROM clients WHERE name LIKE '%${q}%'`)`

FIX

Use parameterized queries: db.$queryRaw`... WHERE name LIKE ${'%' + q + '%'}`. Never interpolate user input into a raw SQL string.

HIGH

No spend cap on the AI summary endpoint

COST-03 · API cost & LLM spend control · high confidence

› `app/api/summarize/route.ts:9 — calls the model on every request with no per-user or global budget check`

› `No usage table, counter, or provider hard-limit found in the repo`

FIX

Add a per-user monthly token budget enforced in-app before the call, plus a hard billing cap and alert thresholds in the provider console.

HIGH

No rate limiting on login or password-reset

AUTH-05 · Authentication & access control · high confidence

› `app/api/auth/login/route.ts:1 — handler has no rate-limit middleware`

› `No upstash/ratelimit, no middleware.ts matcher for /api/auth/*`

FIX

Add IP + account rate limiting and lockout/backoff to login, reset, and token endpoints (e.g. @upstash/ratelimit).

HIGH

Full request bodies (including card metadata) written to logs

LOG-05 · Logging & observability · medium confidence

› `middleware.ts:14 — console.log('REQ', req.method, req.url, await req.text())`

FIX

Remove body logging on payment routes; adopt structured logging with an explicit allowlist of fields and redaction of tokens/PII.

HIGH

Unhandled promise rejection crashes the webhook handler on malformed input

TEST-03 · Testing beyond the happy path · medium confidence

› `app/api/webhooks/stripe/route.ts:7 — JSON.parse(rawBody) with no try/catch; a non-JSON body returns a 500 and drops the event`

FIX

Wrap parsing/verification in try/catch, verify the Stripe signature first, return 400 on bad input, and make the handler idempotent.

HIGH

Two dependencies carry known high-severity advisories

DEP-01 · Dependencies & supply chain · high confidence

- `package-lock.json — next@14.1.0 (advisory GHSA-… SSRF in image optimizer)`
- `package-lock.json — axios@0.21.1 (advisory GHSA-… SSRF/redirect)`

FIX

Upgrade next and axios to patched releases; add Dependabot or npm audit to CI to catch future advisories.

HIGH

PDF generation runs synchronously in the request path

SCL-01 · Scaling & performance · medium confidence

- `app/api/invoices/[id]/pdf/route.ts:20 — renders the PDF inline and can take 4–8s under load, holding the request open`

FIX

Move PDF generation to a background job/queue and return a job id; deliver the finished PDF via signed URL or webhook.

HIGH

No error tracking integrated

LOG-02 · Logging & observability · high confidence

- `No Sentry/Bugsnag/OpenTelemetry SDK in package.json; errors are console.error only and lost on Fly log rotation`

FIX

Integrate an error-tracking service and report exceptions centrally with request/user context.

HIGH

No CI pipeline runs tests or builds before deploy

DEP-REL-02 · Deployment & release · high confidence

- `No .github/workflows or other CI config; README says deploy with \`fly deploy\` from a laptop`

FIX

Add a CI pipeline that runs the test suite and build on every PR and gates deploys on green.

HIGH

No privacy policy or data-use disclosure shipped

CMP-01 · Compliance & privacy · medium confidence

- `No /privacy or /legal route; no privacy copy in the repo, though the app stores client names, emails, and invoice amounts (PII)`

FIX

Publish an accurate privacy policy and data-use disclosure; document what PII is collected and add an export/delete path.

Needs review & access needed

Items that could not be confirmed from what was provided. Give the audit the artifact named on each to close it — an unverified item is never a silent pass.

NEEDS REVIEW

Automated backups and a tested restore could not be confirmed

DB-02 · critical

No backup schedule, snapshot policy, or restore runbook is present in the repo. The database is managed on Fly Postgres, whose backup settings live outside the codebase.

PROVIDE TO CLOSE

Confirmation that Fly Postgres automated snapshots are enabled (frequency + retention) and evidence of at least one successful test restore.

NEEDS REVIEW

Infrastructure is not captured as version-controlled code

INF-01 · high

fly.toml defines the app process but there is no IaC for the database, secrets, networking, or DNS. Unclear whether the rest is click-ops.

PROVIDE TO CLOSE

The Terraform/Pulumi config (or confirmation the remaining infra is click-ops), plus the Fly org network/firewall settings.

NEEDS REVIEW

Database public exposure could not be verified

INF-06 · high

Whether Postgres is reachable only over the private Fly network or has a public endpoint cannot be determined from the repo.

PROVIDE TO CLOSE

The Fly Postgres attach/network configuration showing the DB is on the private 6PN network with no public IP.

NEEDS REVIEW

Test suite covers only the happy path of two routes

TEST-01 · high

A vitest suite exists (tests/) but exercises only successful invoice create + list. No error, auth-boundary, or concurrency cases.

PROVIDE TO CLOSE

Confirmation of intended coverage targets; the audit recommends boundary + failure-path tests before launch regardless.

NEEDS REVIEW

No feature-flag / kill-switch mechanism for the AI feature

FLAG-01 · medium

The AI summary feature ships all-or-nothing; no flag to disable it without a redeploy.

PROVIDE TO CLOSE

Confirmation of whether a kill-switch is required for launch; recommended for the AI path given COST-03.

Per-domain detail

Every check, grouped by doctrine domain. Each domain notes the matching "Before You Go Live" doctrine numbers for the full failure mode and the fix.

Authentication & access control

Doctrine 01–03

FAIL	Invoice queries are not scoped to the caller's tenant	AUTH-03
FAIL	No rate limiting on login or password-reset	AUTH-05
PASS	Identity delegated to a managed provider	AUTH-01
PASS	No plaintext or weak password storage	AUTH-04

Secrets & credentials

Doctrine 04

FAIL	Live Stripe secret key committed to the repository	SEC-01
PASS	Most secrets loaded from environment, not hardcoded	SEC-04

Input validation & web vulnerabilities

Doctrine 05

FAIL	Search endpoint builds SQL by string concatenation	INP-03
PASS	Output is escaped by default; no unsafe HTML injection	INP-02

Dependencies & supply chain

Doctrine 06

FAIL	Two dependencies carry known high-severity advisories	DEP-01
PASS	Locked, reproducible builds	DEP-04

Database hygiene

Doctrine 15

REVIEW	Automated backups and a tested restore could not be confirmed	DB-02
PASS	Schema changes go through versioned migrations	DB-01

Scaling & performance

Doctrine 09–10

FAIL	PDF generation runs synchronously in the request path	SCL-01

API cost & LLM spend control

Doctrine 11–12

FAIL	No spend cap on the AI summary endpoint	COST-03

Logging & observability

Doctrine 13–14

FAIL	Full request bodies (including card metadata) written to logs	LOG-05
FAIL	No error tracking integrated	LOG-02

Deployment & release

Doctrine 16

FAIL	No CI pipeline runs tests or builds before deploy	DEP-REL-02

Feature flags & progressive rollout

Doctrine 17

REVIEW	No feature-flag / kill-switch mechanism for the AI feature	FLAG-01

Infrastructure, containers & networking

Doctrine 07

REVIEW	Infrastructure is not captured as version-controlled code	INF-01
REVIEW	Database public exposure could not be verified	INF-06

Testing beyond the happy path

Doctrine 18

FAIL	Unhandled promise rejection crashes the webhook handler on malformed input	TEST-03
REVIEW	Test suite covers only the happy path of two routes	TEST-01

Compliance & privacy

Doctrine 22

FAIL	No privacy policy or data-use disclosure shipped	CMP-01

Documentation & handoff

Doctrine 23

PASS	README lets a new engineer run the app	DOC-01

Closing checklist

The go-live rule for each domain, marked against this build. ☑ clear · ▨ needs review · ☒ blocked. Tick each blocked and review item off, re-audit, then ship.

- ☒ **Authentication & access control.** blocked — see launch blockers
- ☒ **Secrets & credentials.** blocked — see launch blockers
- ☒ **Input validation & web vulnerabilities.** blocked — see launch blockers
- ☒ **Dependencies & supply chain.** blocked — see launch blockers
- ▨ **Database hygiene.** needs review before ship
- ☒ **Scaling & performance.** blocked — see launch blockers
- ☒ **API cost & LLM spend control.** blocked — see launch blockers
- ☒ **Logging & observability.** blocked — see launch blockers
- ☒ **Deployment & release.** blocked — see launch blockers
- ▨ **Feature flags & progressive rollout.** needs review before ship
- ▨ **Infrastructure, containers & networking.** needs review before ship
- ☒ **Testing beyond the happy path.** blocked — see launch blockers
- ☒ **Compliance & privacy.** blocked — see launch blockers
- ☐ **Documentation & handoff.** clear

This is a SAMPLE report for a fictional application ("Ember Ledger") created to show the format and depth of a Launchproof production-readiness audit. It is not a review of a real product and describes no real company, codebase, or security posture. Every finding, file path, and line number below is illustrative. Your audit is run against your own repository and its findings are entirely your own.