AI MVP Production Readiness Checklist.

54 checks across nine areas where AI-generated MVPs fail in production. Print this, work through it with your team or hand it to whoever owns the platform.

01 Security & Secrets

No hard-coded credentials. No keys in the client bundle. Real rate limits.

- [ ] No secrets in source code or git history
- [ ] No service-role or admin keys exposed in client bundles
- [ ] Rate limits on every authenticated endpoint
- [ ] Rate limits on every unauthenticated endpoint (including auth itself)
- [ ] Dependencies scanned for known CVEs
- [ ] Container images scanned and rebuilt on a schedule
- [ ] CORS configured (not * on production)
- [ ] Security headers set (CSP, X-Frame-Options, HSTS, Referrer-Policy)

02 Auth & Access Control

Authorization decisions made on the server, not the client.

- [ ] Session tokens stored securely (httpOnly cookies or equivalent)
- [ ] Password reset and account recovery flows tested end-to-end
- [ ] Role and permission checks enforced server-side (never trust client)
- [ ] Multi-factor auth available for admin / privileged users
- [ ] Account lockout / brute-force protection on login
- [ ] Audit log for sensitive actions (role change, billing, data export)

03 Architecture

Intentional boundaries - not whatever the AI suggested first.

- [ ] Service boundaries are documented and intentional
- [ ] No business logic in the client
- [ ] Critical write paths have idempotency keys
- [ ] Background jobs survive worker restarts
- [ ] Concurrency model is explicit (no hidden race conditions)
- [ ] External API contracts are versioned

04 Infrastructure

Reproducible, least-privilege, with backups that have been restored at least once.

- [ ] Infrastructure is in code (Terraform / Pulumi / CDK - not the cloud console)
- [ ] Production, staging and dev environments are separated
- [ ] IAM follows least-privilege (no broad admin policies in production)
- [ ] Storage buckets are private by default
- [ ] Backups are automated AND tested by restoring
- [ ] Secrets live in a real secret manager, not env files in git

05 CI / CD

Deploy on a Tuesday, roll back on a Tuesday, sleep on a Tuesday.

- [ ] Production deploys are automated (no copying files to a server)
- [ ] Every deploy can be rolled back in under 5 minutes
- [ ] Staging uses the same image as production
- [ ] Database migrations are versioned and applied automatically
- [ ] Migrations are tested on a copy of production data
- [ ] CI runs the full test suite before merge
- [ ] Secrets are not exposed in build logs or artifacts

06 Observability

You find issues before your customers do.

- [ ] Structured logs (JSON) with consistent fields and a request ID
- [ ] Request IDs propagate across services
- [ ] Metrics cover the four golden signals (latency, traffic, errors, saturation)
- [ ] Distributed tracing for the critical path
- [ ] Alerts wake up a real human when production breaks
- [ ] Alerts are tuned - no daily false positives
- [ ] An external uptime check monitors a real user flow, not just the index page

07 Database & Scaling

The query plan you assume is the query plan you have.

- [ ] Indexes on every column used in a WHERE clause on a hot path
- [ ] No N+1 queries on critical endpoints
- [ ] Connection pooling configured (not a new connection per request)
- [ ] Read replicas in place if the workload is read-heavy
- [ ] Cache layer with explicit invalidation rules
- [ ] A realistic load test has been run against staging

08 Operational Readiness

Someone is responsible when the pager goes off - and they know what to do.

- [ ] Someone is on call for production
- [ ] Incident runbook exists (even a one-page doc counts)
- [ ] Customer support has a way to reach engineering
- [ ] Status page or equivalent communication channel
- [ ] Postmortems written for production incidents
- [ ] On-call rotation is sustainable (not one founder, always)

09 AI-Generated Code Risks

AI accelerates writing code. It does not accelerate reading code.

- [ ] Every AI-generated file has been read by a human
- [ ] No secrets copy-pasted from AI suggestions or example snippets
- [ ] AI-suggested dependencies exist and are maintained (no hallucinated packages)
- [ ] AI-generated SQL has been reviewed for injection risk
- [ ] AI-generated infrastructure code has been reviewed for over-permissive IAM
- [ ] AI-generated tests actually test the intended behaviour, not just the happy path