FREE DIAGNOSTIC

The 27-Point AI App Production Readiness Checklist

A practical founder-and-engineer review for applications built quickly with AI-assisted tools.

What this checklist does

It exposes the production risks that rapid prototyping commonly hides: authorization gaps, unsafe model actions, duplicate side effects, missing recovery, provider failure, runaway cost, and blind deployments.

Who should use it

Founders approaching launch You have a working product and need to know what must be fixed before real customers, payments or sensitive data arrive.	**Teams inheriting AI-built code** The application works, but no senior engineer has validated the architecture, failure behavior or operating controls.
Agencies handing off an MVP You need a defensible readiness review before the client treats a fast prototype as a production system.	**Technical leaders under pressure** You need a risk-ranked plan instead of an open-ended rewrite proposal.

How to score

Score	Meaning	Use this test
0	Missing / unknown	There is no evidence, owner or reliable control.
1	Partial / fragile	A control exists, but it is manual, incomplete, untested or easy to bypass.
2	Production-ready	The control is explicit, tested, monitored and owned.

Maximum score: 54. Scores are directional, not a compliance certificate. A single zero on authentication, authorization, backup restore, payment integrity or high-impact AI actions can outweigh a respectable total.

1 Identity, access and permissions

Score each control 0, 1 or 2 and record the evidence you actually observed.

01. Authentication is enforced

SCORE
[0][1][2]

What good looks like: Every protected action verifies a real user or trusted service identity. There is no client-side-only protection.

Evidence / gap / owner:

02. Authorization is explicit

SCORE
[0][1][2]

What good looks like: Roles, ownership and tenant boundaries are checked server-side for every sensitive read and write.

Evidence / gap / owner:

03. Privileged access is controlled

SCORE
[0][1][2]

What good looks like: Admin paths use least privilege, stronger authentication, and a traceable break-glass process.

Evidence / gap / owner:

2 Secrets and supply-chain safety

Score each control 0, 1 or 2 and record the evidence you actually observed.

04. Secrets never live in source code

SCORE
[0][1][2]

What good looks like: API keys, tokens and credentials are stored in a secret manager or protected deployment environment.

Evidence / gap / owner:

05. Dependencies are reviewed and pinned

SCORE
[0][1][2]

What good looks like: Production builds are reproducible; vulnerable or abandoned packages are identified and handled.

Evidence / gap / owner:

06. Third-party access is scoped

SCORE
[0][1][2]

What good looks like: External providers receive only the permissions and data required for the exact workflow.

Evidence / gap / owner:

3 Data protection and privacy

Score each control 0, 1 or 2 and record the evidence you actually observed.

07. Sensitive data is classified

SCORE
[0] [1] [2]

What good looks like: The team knows what personal, financial, health or confidential data exists and where it moves.

Evidence / gap / owner:

08. Encryption and retention are defined

SCORE
[0] [1] [2]

What good looks like: Sensitive data is protected in transit and at rest, and is not retained indefinitely by accident.

Evidence / gap / owner:

09. Deletion and export are possible

SCORE
[0] [1] [2]

What good looks like: Customer data can be found, exported and deleted without manual database archaeology.

Evidence / gap / owner:

4 Database integrity and recovery

Score each control 0, 1 or 2 and record the evidence you actually observed.

10. Schema constraints protect correctness

SCORE [0] [1] [2]

What good looks like: Uniqueness, foreign keys and validation rules prevent invalid state, not only UI checks.

Evidence / gap / owner:

11. Critical writes are atomic and idempotent

SCORE [0] [1] [2]

What good looks like: Retries cannot create duplicate orders, charges, messages or side effects.

Evidence / gap / owner:

12. Backups are restore-tested

SCORE [0] [1] [2]

What good looks like: Backups exist, restore procedures are documented, and a restoration has been tested recently.

Evidence / gap / owner:

5 API and integration resilience

Score each control 0, 1 or 2 and record the evidence you actually observed.

13. Inputs are validated at trust boundaries

SCORE [0] [1] [2]

What good looks like: Payloads, file uploads and provider callbacks are validated before they touch core logic.

Evidence / gap / owner:

14. Timeouts, retries and circuit breakers exist

SCORE [0] [1] [2]

What good looks like: External failures are bounded; retries use backoff and do not amplify incidents.

Evidence / gap / owner:

15. Webhooks are authenticated and deduplicated

SCORE [0] [1] [2]

What good looks like: Signatures are verified and repeated deliveries are safe.

Evidence / gap / owner:

6 AI-model controls

Score each control 0, 1 or 2 and record the evidence you actually observed.

16. Model inputs and outputs are treated as untrusted

SCORE [0][1][2]

What good looks like: Prompt injection, unsafe tool calls, malformed output and data leakage are explicitly handled.

Evidence / gap / owner:

17. Cost and latency have hard limits

SCORE [0][1][2]

What good looks like: Token, request, timeout and concurrency budgets prevent one workflow from creating an uncontrolled bill or outage.

Evidence / gap / owner:

18. Fallback and human approval paths exist

SCORE [0][1][2]

What good looks like: High-impact actions require approval or deterministic checks, and the system degrades safely when models fail.

Evidence / gap / owner:

7 Observability and incident response

Score each control 0, 1 or 2 and record the evidence you actually observed.

19. Logs are structured and useful

SCORE
[0] [1] [2]

What good looks like: Important requests and decisions can be traced without exposing secrets or unnecessary personal data.

Evidence / gap / owner:

20. Metrics and alerts cover customer impact

SCORE
[0] [1] [2]

What good looks like: The team sees error rate, latency, queue depth, provider failure, saturation and business-critical failures.

Evidence / gap / owner:

21. Incident ownership is clear

SCORE
[0] [1] [2]

What good looks like: Someone is responsible, severity levels are defined, and the first response steps are documented.

Evidence / gap / owner:

8 Deployment and change safety

Score each control 0, 1 or 2 and record the evidence you actually observed.

22. Production changes are reproducible

SCORE [0] [1] [2]

What good looks like: Deployments come from versioned code and controlled configuration, not one-off manual edits.

Evidence / gap / owner:

23. Rollback is fast and tested

SCORE [0] [1] [2]

What good looks like: A broken release can be reversed without improvisation or destructive database changes.

Evidence / gap / owner:

24. Environments and access are separated

SCORE [0] [1] [2]

What good looks like: Development and test systems cannot silently mutate production data or use production credentials.

Evidence / gap / owner:

9 Capacity, performance and business continuity

Score each control 0, 1 or 2 and record the evidence you actually observed.

25. Realistic load has been tested

SCORE
[0] [1] [2]

What good looks like: The system has been exercised under expected peak concurrency, payload size and background work.

Evidence / gap / owner:

26. Bottlenecks have protective limits

SCORE
[0] [1] [2]

What good looks like: Queues, rate limits, pagination, caching and resource caps prevent cascading failure.

Evidence / gap / owner:

27. Critical dependencies have a continuity plan

SCORE
[0] [1] [2]

What good looks like: The team knows what happens when hosting, payment, email, AI or another key provider is unavailable.

Evidence / gap / owner:

DECISION PAGE

Readiness Score and Top-Five Risks

Convert observations into an owner-based remediation plan.

Category	Score	Max
Identity, access and permissions		6
Secrets and supply-chain safety		6
Data protection and privacy		6
Database integrity and recovery		6
API and integration resilience		6
AI-model controls		6
Observability and incident response		6
Deployment and change safety		6
Capacity, performance and business continuity		6

TOTAL: _____ / 54

> **Interpretation**
> 43-54: generally strong, investigate every zero. 30-42: material production risk; schedule remediation before scale. 0-29: do not treat the application as production-ready without focused hardening.

Top-five risks

#	Risk / failure scenario	Severity	Owner	Target date
1				
2				
3				
4				
5				

INDEPENDENT ASSESSMENT

Turn the Checklist into a Decision

A checklist exposes uncertainty. The audit converts it into evidence, priorities, ownership and verification.

72-Hour Production Readiness Audit - ILS 4,900

A fixed-scope review of architecture, data, security, deployment, observability, AI controls and failure behavior. You receive a risk-ranked report, 30-day plan and recorded walkthrough.

The audit is useful when

Launch is approaching Real users, payments, sensitive data or a customer commitment make hidden failure expensive.	**The codebase was built rapidly** AI-assisted tools or multiple contributors produced speed, but no senior owner has validated the whole system.
A handoff or diligence event is coming A buyer, investor, client or internal team needs an evidence-based view of the system.	**The team is debating a rewrite** A risk-ranked assessment prevents an expensive architecture project before the actual blockers are known.

What you receive

- Executive decision summary and readiness scorecard.
- Prioritized findings with evidence, failure scenario and recommended action.
- A 30-day remediation plan with owners and verification tests.
- One recorded 60-minute walkthrough and Q&A.

Start with a 20-minute fit call

Email ilan.melki@gmail.com or visit rescue.asimtop.com. The audit is offered only when the scope, evidence and business event make it useful.