R RepoRisk Receipt

SAMPLE DELIVERABLE

RepoRisk Launch Risk Receipt

A realistic mock report for an AI-built SaaS before live users, payments or private customer data are exposed.

LAUNCH RISK SCORE

72 / 100

3	6	8
HIGH RISK	MEDIUM	LOW RISK

Project
ai-saas-demo

Receipt ID
RR-SAMPLE-7249

Stack
Next.js / Stripe / Supabase / Clerk / Vercel

Recommendation
Fix before paid launch

Top 3 Launch Blockers

1. Stripe webhook signature verification missing
2. Supabase RLS policy not detected
3. Admin API route lacks server-side role check

Top 3 Launch Blockers

1

High **Stripe webhook signature verification missing**

`app/api/webhooks/stripe/route.ts`

Why it matters: Fake or replayed webhook events may update subscription status without a real Stripe event.

Fix: Read the raw request body and verify Stripe-Signature with STRIPE_WEBHOOK_SECRET before mutating payment state.

2

High **Supabase RLS policy not detected**

`supabase/migrations/2026_create_tables.sql`

Why it matters: Authenticated users may read or write records that belong to another workspace.

Fix: Enable RLS on customer-owned tables and add owner or organization-scoped policies.

3

High **Admin API route lacks server-side role check**

`app/api/admin/users/route.ts`

Why it matters: Hiding admin controls in the UI is not authorization. Direct API calls can still reach the route.

Fix: Validate the server session and require an admin role inside the route handler.

Medium Risk Findings

- Checkout success page is treated as payment confirmation.
- stripeSubscriptionId does not have a unique database constraint.
- Billing event payloads use broad any types.
- Empty catch block hides failed project writes.
- Missing idempotency guard for repeated provider events.
- Account billing panel does not show failed payment state.

Low Risk Findings

- TODO remains in the billing retry path.
- Environment variables are not documented in README.
- Preview deployment uses production-looking copy.
- Admin user search has no smoke test.
- Webhook error logs are not structured.
- Customer portal link lacks a loading state.
- Duplicate validation logic appears in two API routes.
- No launch checklist exists in the repository.

Recommended Fix Order

1. Block unverified Stripe webhooks before any payment state writes.
2. Enable Supabase RLS and owner-scoped policies on customer tables.
3. Add server-side admin role checks to every admin API route.
4. Add unique constraints for Stripe customer and subscription ids.
5. Add regression tests for checkout, webhook, auth and admin paths.

Cursor / Claude Code Repair Prompts

Stripe webhook repair prompt

```
Update app/api/webhooks/stripe/route.ts so the handler reads the raw body, verifies Stripe-Signature with STRIPE_WEBHOOK_SECRET, rejects invalid signatures, and only then updates subscription state. Add a test or fixture proving unsigned payloads are rejected.
```

Supabase RLS repair prompt

```
Inspect supabase/migrations/2026_create_tables.sql. Enable RLS for customer-owned tables and add owner-scoped select, insert and update policies using auth.uid() or organization membership. Include a migration note listing protected tables.
```

Admin route repair prompt

```
Review app/api/admin/users/route.ts. Add server-side session validation and require an admin role before returning data or mutating users. Return 401 for missing sessions and 403 for non-admin users.
```

Launch Recommendation

Do not launch paid users before fixing these 3 issues. Re-run the receipt after the high-risk payment, RLS and admin authorization paths are repaired.