

Cenk Kurtoglu

Next.js / TypeScript / Supabase / Vercel - Technical Capability Brief

A concise, evidence-based profile for scoped product engineering, Supabase security reviews and production debugging. This document is not an employment-history resume; it deliberately avoids unverified experience claims.

cenkkurtoglu.com	github.com/cekuu35	LinkedIn
---	---	--------------------------

CORE CAPABILITIES

Product engineering	Next.js App Router, React, TypeScript, responsive UI, server rendering, route handlers and Vercel delivery.
Data and authorization	Supabase/PostgreSQL schemas, authentication boundaries, Row Level Security, cross-user isolation tests and remediation notes.
Reliability	Regression-focused debugging, acceptance criteria, webhook replay safety, failure-path analysis and production verification.
Technical communication	Bounded tasks, reproducible fixtures, concise audit findings, explicit limitations and review-ready documentation.

PUBLIC, INSPECTABLE EVIDENCE

Supabase RLS leak fixture - Broken and fixed branches demonstrate forbidden cross-user read/write cases and a five-test isolation matrix.

<https://github.com/cekuu35/supabase-rls-leak-demo>

Membership SaaS architecture - Reference design for database-enforced entitlement, idempotent billing events and authorization verification. Clearly labeled as a reference, not a client deployment.

<https://cenkkurtoglu.com/membership-saas-architecture>

Supabase RLS audit workflow - Public service scope, sample audit structure, safety boundaries and fixed-price contracting route.

<https://cenkkurtoglu.com/supabase-rls-security-audit>

Engineering task portfolio - Examples of converting ambiguous failures into reproducible tasks, acceptance checks and explicit limitations.

<https://cenkkurtoglu.com/software-engineering-task-author>

WORKING APPROACH

1. Define the user roles, protected data and failure invariants before implementation.
2. Ship one end-to-end workflow with authorization and regression checks before expanding scope.
3. Keep billing and other critical state deterministic; AI may assist workflows but does not invent or mutate financial truth.
4. Verify production behavior, document known limits and hand over reviewable evidence.

BEST FIT	ENGAGEMENT NOTE
Next.js and Supabase feature work; RLS audits; production debugging; Vercel deployment; scoped technical writing.	Availability, rate, timeline and access requirements are confirmed separately for each engagement. No production secrets should be sent before scope and access are agreed.