

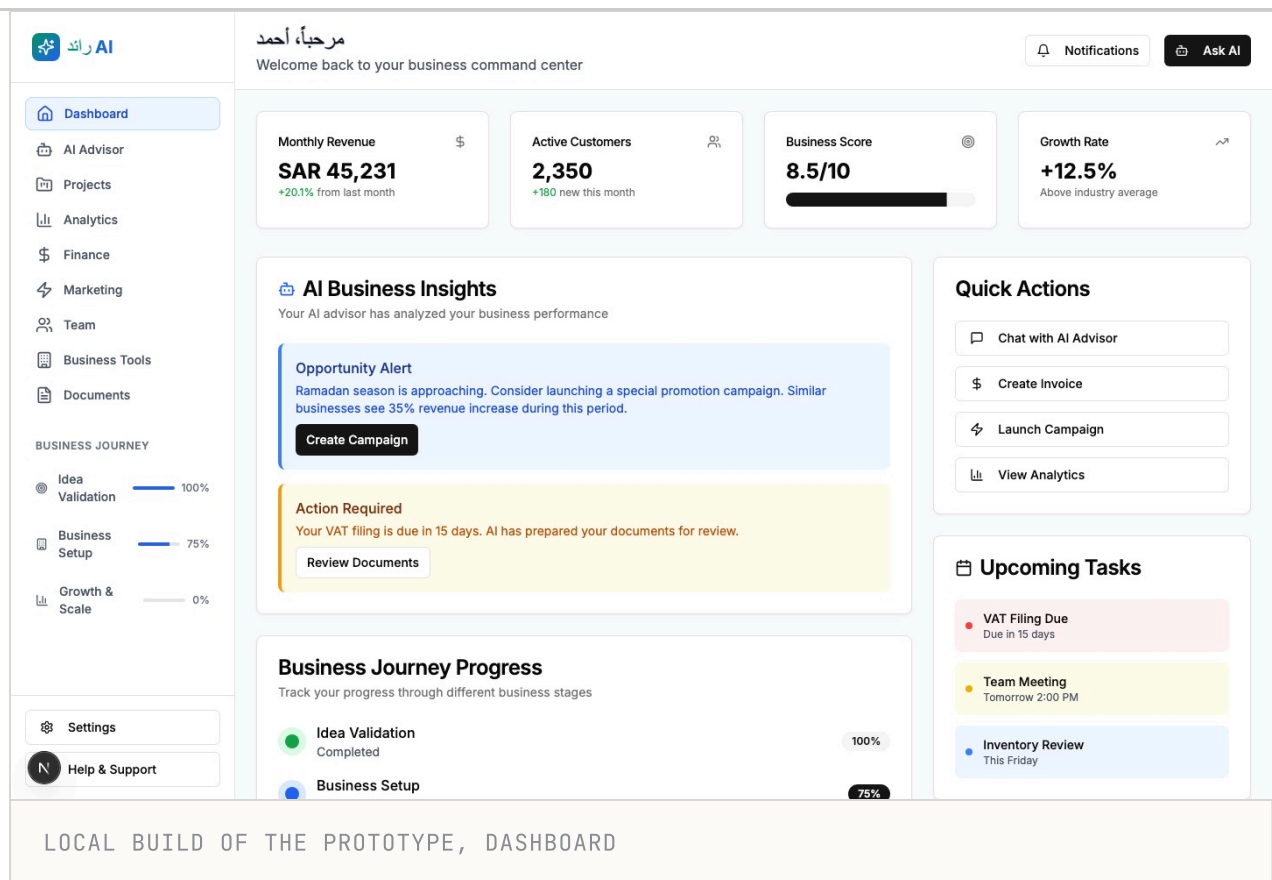
Saudi Entrepreneur OS: a business-operating-system prototype built with AI-assisted UI generation

Saudi Arabia's small-business boom created a gap: founders juggling company setup, Zakat and VAT filings, invoicing, hiring and marketing across a dozen disconnected tools. I wanted to test whether a single "business operating...

Role: Product owner and prototyper

Stack: Next.js (App Router), TypeScript, Tailwind, shadcn/ui and Radix primitives, react-hook-form; interface generated and iterated with Vercel's v0, then synced to GitHub and deployed on Vercel

Status: Interactive prototype with static data. No backend. Two repositories hold the same prototype (`saudi-entrepreneur-app` and the v0-synced `ksa-ai-dev`); this page covers both.



Context

Saudi Arabia's small-business boom created a gap: founders juggling company setup, Zakat and VAT filings, invoicing, hiring and marketing across a dozen disconnected tools. I wanted to test whether a single "business operating system" for a Saudi founder would make sense before investing engineering time, and to test something else at the same time: how far AI-assisted UI generation could carry a full product prototype in days rather than weeks.

What I built

A complete product surface, roughly forty screens

- ◆ **Stages.** Idea, setup and growth tracks that shape which modules a founder sees first.
- ◆ **Finance.** Invoices, expenses, reports, VAT, Zakat and a tax-filing flow, modelled on the actual Saudi obligations rather than generic accounting screens.
- ◆ **Projects.** Kanban, Gantt, calendar, reports and settings.
- ◆ **Team.** Attendance and people management.
- ◆ **Marketing and analytics.** Campaign creation and customer analytics.
- ◆ **Documents, help and an AI chat surface** for the assistant that would sit across the product.

The method

The interface was produced through v0 from written specifications, screen by screen, with the generated code synced to the repository and deployed on Vercel. My work was the product thinking: the module map, the Saudi-specific finance flows, the stage-based onboarding, and the review of what the generator produced. Where the output was generic, the specification was rewritten until it was not.

Decisions worth noting

- ◆ **Prototype the whole product, not one screen.** A founder evaluates a business OS by breadth. Forty coherent screens told me far more in stakeholder conversations than a polished dashboard would have.
- ◆ **Localise the finance model first.** Zakat and VAT as first-class modules, not a "taxes" placeholder, is what made the demo credible to Saudi founders.
- ◆ **Use the generator for what it is good at.** Layout and component scaffolding came from the tool; the domain model, information architecture and copy did not. That split is the honest lesson of the exercise, and it is how I now scope AI-assisted builds for clients.

Why it is in the portfolio

It is presented at its real size: a prototype, not a product. It earns its place because it demonstrates two things a CTO-as-a-service client cares about: product scoping for a specific regulated market, and a clear-eyed, tested view of where AI-assisted development helps and where it does not.