

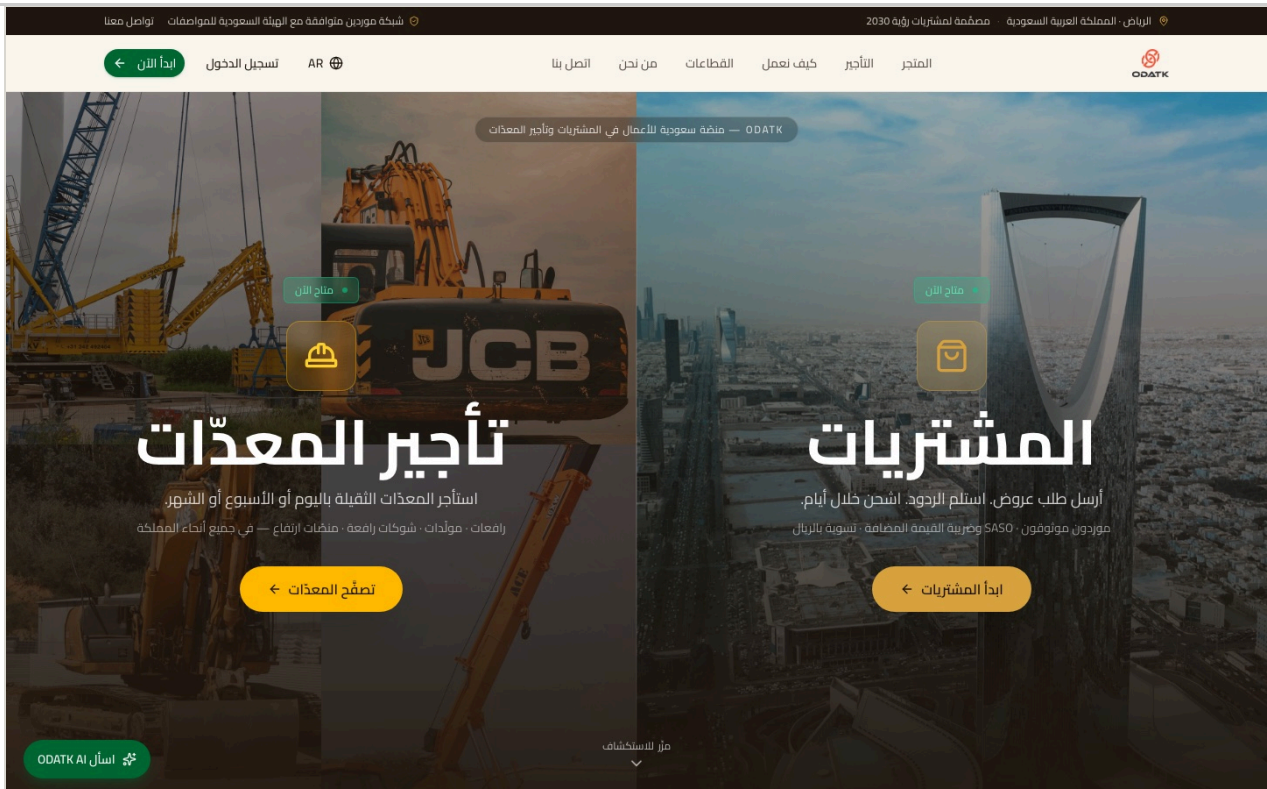
ODATK: a Saudi B2B procurement and equipment-rental marketplace

ODATK is a two-sided B2B marketplace for Saudi Arabia, built for the procurement patterns of construction, oil and gas, manufacturing and facility management. Buyers publish requests for quotation, suppliers respond with quotes...

Role: Founder, sole architect and developer

Stack: Next.js 16 (frontend), Express (API), MongoDB, JWT in an HttpOnly cookie, Vercel AI SDK with OpenAI GPT-4o-mini, Arabic-first with English

Live: odatk.com



Context

ODATK is a two-sided B2B marketplace for Saudi Arabia, built for the procurement patterns of construction, oil and gas, manufacturing and facility management. Buyers publish requests for quotation, suppliers respond with quotes, buyers compare and award, and the order is tracked through fulfilment. A second surface rents heavy equipment (cranes, generators, lifts, excavators) by the day, week or month, from a supplier network positioned around SASO conformity and Saudi Vision 2030 procurement. The live site is Arabic-first and right to left, with a split hero that sends a visitor either to procurement or to rental in one click.

I built it alone: product, frontend, API, data model, AI features and deployment.

What I built

Three portals on one auth system

Authentication is a JWT held in an HttpOnly cookie. The role on the token drives the post-login redirect, so a client, a supplier and an administrator each land in their own portal.

- ◆ **Client portal.** Create an RFQ, receive supplier quotations, compare them side by side, award, then track the order to delivery.
- ◆ **Supplier portal.** Browse open RFQs, submit quotations, fulfil awarded orders, manage the catalogue.
- ◆ **Admin portal.** Oversight of users, RFQs, orders, content and platform settings.

A public site carries the marketing pages (home, about, services, how it works, industries, contact) with separate registration funnels for buyers and for becoming a supplier.

AI where it earns its place

An "Ask ODATK AI" assistant is present on the live site, and AI features inside the portals run through the Vercel AI SDK against GPT-4o-mini. In the platform's own documentation every feature is typed as static, dynamic, functional or AI, so a reviewer always knows which behaviour they are testing and where the data comes from.

How I kept a solo build honest

Before asking anyone else to look at it I wrote a review packet: a single spreadsheet enumerating every route, page, feature, data source and required role across the whole platform, more than a hundred rows across the public site, auth, the three portals, the API and cross-cutting concerns, plus a seeder that creates the three reviewer accounts. Reviewers work through it row by row and record status, severity and a suggested fix. It is the same discipline I later used for client UAT on the Nomad Universe work: enumerate the surface, then test it systematically.

Decisions and known debt

- ◆ Locale-aware routing was built into the frontend from the start rather than added later; the live product is Arabic-first because the market is.
- ◆ File uploads are currently stored as base64 in MongoDB. I flagged this in the platform's own review notes as the next thing to move to object storage, rather than leaving it as an unstated gap. Knowing exactly where the debt is matters more than pretending there is none.

Outcomes and evidence

- ◆ Live and publicly reachable at odatk.com, Arabic-first, with both the procurement and rental surfaces visible on the home page.
- ◆ A real multi-sided marketplace: three authenticated roles with materially different workflows, not a CRUD demo.
- ◆ Self-authored QA documentation covering the entire surface, written before anyone else tested it.

Screenshot on this page is the live site.