

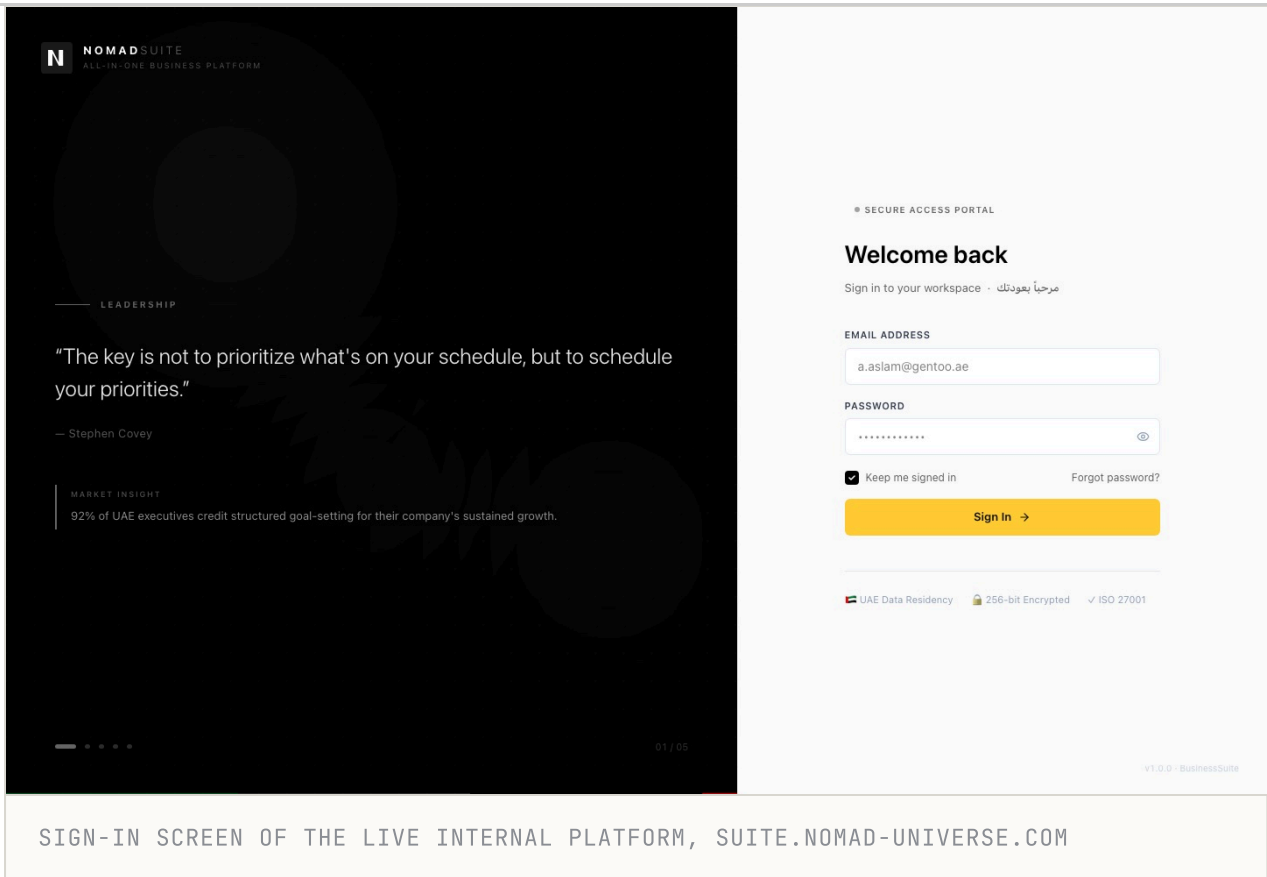
NomadSuite: the internal platform every company in the group runs on

A group of small companies shares people, inventory and a sales pipeline, and none of the off-the-shelf tools fit all three without paying for three separate products and reconciling them by hand. NomadSuite is the internal...

Role: Sole architect and developer

Stack: Next.js (App Router) frontend, NestJS-style backend modules on Prisma and PostgreSQL, role-based access control, multi-company data model, Claude for the design-generation features

Status: In use across the operating companies of the Nomad Universe group



SIGN-IN SCREEN OF THE LIVE INTERNAL PLATFORM, SUITE.NOMAD-UNIVERSE.COM

Context

A group of small companies shares people, inventory and a sales pipeline, and none of the off-the-shelf tools fit all three without paying for three separate products and reconciling them by hand. NomadSuite is the internal platform I built to replace that: one login, one company switcher, and modules for HR and leave, inventory, and bid-and-proposal work, with role-based permissions per company.

What I built

Multi-company foundation

A `Company` and `UserCompany` model with its own RBAC layer (roles and members), so one person can belong to several companies with different permissions in each and switch between them without logging out. Dashboard preferences are stored per user. New-account requests go through an approval flow rather than open sign-up.

HR and leave

Employees, departments, leave types, public holidays and per-company leave settings as first-class data. The interface is built around the way people actually ask about leave: a Wall Chart showing who is out when across a team, an approvals queue, an activity log, and a per-employee "My Year" view that shows an individual their entitlement, what they have used and what is booked. Entitlements are computed, not typed into a spreadsheet.

Inventory and warehouse

A live folder tree of inventory (folders, items, tags, assets), item-level activity history, search, tagging, workflows for moving stock through states, and reports including low-stock views. It replaced a third-party inventory subscription with something the group owns.

BPS: bid and proposal, with AI doing the creative first draft

This is the module that surprised people. The events business responds to RFPs that need a staging and décor concept, quickly. BPS takes the RFP (event name, client, theme, constraints) and uses Claude to generate a spatial stage-design concept together with a complete image-generation render prompt, and separately generates décor concepts for table setups and centrepieces, extracting a colour palette from each. Everything a proposal produces (the RFP analysis, saved concepts, stage designs, graphics prompts) lands in one activity feed, and any item can be exported as Markdown to drop straight into the proposal document. It turns the slowest part of a bid into a starting point that a designer refines rather than a blank page.

What is designed but not yet shipped

The platform's subscription and billing layer, and scheduled report subscriptions (low-stock reports on a cadence, delivered by email as CSV or PDF), exist as designs: a subscription-management interface including the cancellation state, and reference mockups for the report-subscription flow. The backend for Stripe billing and the report cron are the next build phase. I say so plainly here because an earlier version of my resume described them as delivered, and when I checked the code they were not.

Decisions worth noting

- **One data model, several products.** HR, inventory and BPS are modules on the same company and permission model rather than three apps with three user tables, which is what makes the company switcher and the shared RBAC possible.
- **AI for the first draft, humans for the decision.** BPS generates render prompts and concepts; it does not pick the winning design. That boundary is deliberate.
- **Replace subscriptions with owned software only where the fit is bad.** The inventory module replaced a paid tool because the group's workflow did not match it. Where an external tool fits, we keep it.

Evidence

The repository lives under the Gentoo-Tickets organisation with mirrors under my own accounts. Module structure and data model as described above are the current `main` branch.