TECH STYLE LTD

TECH-STYLE.CO

PRODUCT BROCHURE

Rutba ERP — Product Brochure

The open-source, modular business management system: 22 Next.js applications on one shared API.

WHAT IS INSIDE

Document	Rutba ERP brochure
Generated	2 September 2026
Online	https://www.tech-style.co/docs/rutba-erp-brochure

An open-source, modular business management system. Twenty-two independent Next.js applications share one API, one authentication portal and one component library — so you can run the whole suite or just the modules you need.

Modular by construction, not by configuration

Most ERPs are one enormous application with features switched off. Rutba ERP inverts that: each functional domain is a separate Next.js application in an npm workspaces monorepo, and they cooperate through a shared Strapi 5 API rather than a shared codebase.

That means a retailer can deploy point of sale and stock alone, add order management when they start shipping, and bring in manufacturing or payroll later — without carrying the weight of modules they never open, and without a migration each time.

Shared authentication, shared UI components and a generated API client keep the apps consistent. A descriptor-driven permission layer means a role defined once is enforced everywhere.

Structure	npm workspaces monorepo
Apps	22 independent Next.js applications
API	Strapi 5, with a Core API taking over
Auth	OAuth-style JWT with per-app access

Proved in production.
The commerce modules — storefront, checkout, order pipeline and rider delivery — run Rutba.pk, a live retail store. Edge cases surface there before they reach anyone else.

How this relates to Rutba Suite.
Rutba ERP is the open-source monorepo, MIT licensed and free to run yourself — and it is the generation this work came from. Rutba Suite is the next generation: the same business knowledge rebuilt as SaaS on a new engine, with many more applications than the ERP ever had, a workspace and platform services beside them, and one identity, licensing and billing plane underneath. Not a repackaging — a modernisation. The repository stays open and stays yours to run; the suite is what you subscribe to when you would rather we ran it.

MODULES

Twenty-two applications, one platform

Every module below is a deployable application in its own right, sharing state through the central API.

Commerce & sales

Point of Sale

Sales processing, cart management, returns, cash register operations and financial reports for counter and shop-floor selling.

Storefront

A server-rendered Next.js 16 shop with sale offers, express and full-address checkout, and cash on delivery.

Order Management

Stage-based order processing built around a state-machine chokepoint — returns, label printing, delivery operations, rider assignment and payment verification.

My Orders

A customer-facing portal for order tracking, returns and account management.

Inventory & supply

Stock Management

Products, purchases, suppliers, brands, categories and day-to-day inventory tracking.

Inventory Management

Warehouses and bins, stock levels, transfers, adjustments, cycle counts, batch and expiry tracking, replenishment and inventory valuation.

Reordering engine

MinMax, reorder-point, par-level or manual policies that raise purchase or work orders automatically when stock falls through the floor.

Batch & expiry

FEFO enforcement across serialised, bulk and divisible stock models — so the shortest-dated goods leave first.

Manufacturing

Work orders & operations

Work orders, task and piece-rate tracking, operations sequencing and reusable production templates.

BOM & bundles

Multi-output bills of materials with automatic component consumption, plus bundle definitions for kitted goods.

Material lots & QC

Material issues traced by lot, with quality-control checkpoints recorded against the work order.

Production templates

Repeatable production recipes so a routine build doesn't have to be re-specified every run.

People & finance

HR

Employees, departments, attendance and leave requests.

Employee Self-Service

An employee portal for profile, attendance, leave and payslips — keeping routine requests out of the HR inbox.

Payroll

Salary structures, payroll runs, payslips, deduction rules, employee profiles and adjustments.

Accounting

Chart of accounts, journal entries, invoices, expenses, double-entry posting and fiscal periods.

Customer, content & delivery

CRM

Contacts, leads, activity history and person deduplication so one customer is one record.

Rider App

Delivery offers, active runs, status updates and buyer messaging for the people actually carrying the parcels.

CMS

Page authoring, product groups, CMS sections, SEO defaults and bulk import/export.

Social

Social-account management and a post composer with product preview, for merchandising straight from the catalogue.

Sales channels, support & administration

Marketplace

Channel sync against external marketplaces — listings, category mapping, pricing rules and order pull-back into the same order pipeline.

Helpdesk

Ticketing tied to the customer record and the order it concerns, so support sees the same history sales does.

Campaigns

Audience building and campaign sends over Rutba MTA, keyed to the CRM contacts rather than an exported list.

Mail

Organisation mailboxes and a web client, so the correspondence about an order sits beside the order.

Admin Console

Users, roles, per-app access and the descriptor-driven permission model, administered in one place across every application.

ARCHITECTURE

How the pieces hold together

Independent apps only stay consistent if something enforces it. Three shared packages do that job.

Central Strapi 5 API

One headless API on port 4010 is the single source of truth. Every application talks to it over standard REST endpoints — no app owns another app's data.

Generated API clients

The api-provider package holds one descriptor for every Strapi endpoint, and a scaffolder emits matching client and server bindings per app — so a schema change can't silently drift.

Shared component library

The pos-shared package supplies UI components and context providers, keeping twenty-two applications visually and behaviourally coherent.

Auth portal

An OAuth-style login flow issues JWTs with per-app access rows. Users, roles and app access are administered in one place.

Descriptor-driven permissions

The strapi-api-pro plugin enforces role scope and caches claims at the API, so authorisation is decided once rather than per-frontend.

One environment loader

A central loader with an APP_PREFIX__VAR override convention means each app can be tuned without twenty-two divergent config files.

STACK & REQUIREMENTS

What it runs on

Rutba ERP technical specification

Frontend	Next.js 16 with React 19. Bootstrap 5 for admin and POS interfaces; Tailwind CSS for the storefront.
Backend	Strapi 5.x headless API (port 4010) with a custom strapi-api-pro plugin for descriptor-driven auth, role scope and claim caching, plus a newer Core API (port 4020) that mounts the domain modules directly and is progressively taking the Strapi backend's place.
Database	MySQL 8.x or MariaDB.
Runtime	Node.js 18 or later.
Repository lavout	npm workspaces monorepo — apps/ for applications, packages/ for shared libraries, scripts/ for tooling.
Authentication	OAuth-style JWT flow with per-app app_access rows; X-Rutba-App and X-Rutba-App-Role headers select the active claim set.
Media	Pairs with Rutba Media FileServer as the image origin.
Email	Pairs with Rutba MTA for transactional and campaign delivery.
Licence	MIT — permissive commercial use with attribution.

DEPLOYMENT

Three ways to run it

From a laptop to a container stack to budget shared hosting — the same codebase, three supported paths.

Local development

Start the workspace and each app comes up on its own port — auth on 4003, stock on 4001, sales on 4002, the Strapi API on 4010 and the Core API on 4020, and so on through 4023.

Docker

A full containerised stack — MySQL, Strapi and every Next.js application — for a reproducible production deployment.

Shared hosting

A Passenger + Node.js 22 toolkit that orchestrates build, upload and restart, so the suite runs on inexpensive hosting rather than requiring a cluster.

```
# clone and install the workspace git clone https://github.com/eharain/Rutba-ERP.git cd Rutba-ERP npm
install # bring up the Strapi API, then the apps you need npm run dev --workspace=apps/strapi
```

RELATED

Other products

PACKAGED & SUPPORTED

Rutba Suite

The same work as twenty subscribable products.

RETAIL COMMERCE

Rutba.pk

The commerce modules running a live shop.

MEDIA INFRASTRUCTURE

Media FileServer

The image origin behind the catalogue.

EMAIL DELIVERY

Rutba MTA

Order confirmations and campaigns, paced by reputation.

CONTACT

Talk to us

Tech Style Ltd is a UK technology company building enterprise software, procurement intelligence and AI platforms. Tell us what you are trying to do and we will tell you honestly whether we are the right people for it.

This document online	www.tech-style.co/product-rutba-erp
Website	www.tech-style.co
Enquiries	hello@tech-style.co
Products	www.tech-style.co/products
Partner programme	www.tech-style.co/partners

This document was generated on request.
It is built from the live content of tech-style.co at the moment you downloaded it, so it can never quote a price, a version or a status the site has since moved on from. Request a fresh copy any time at www.tech-style.co/downloads.