



PRINT PRENEUR

BY OKOBOJI GLOBAL PRIVATE LIMITED

The AI agent team behind **print.okoboji.in**

A plain-language guide to what this page is for, the eight agents that answer on it, the technology it runs on, the credentials it needs, and the data it is connected to.

DOCUMENT

System overview

SERVICE

print.okoboji.in

LANGUAGE MODEL

OpenRouter (free tier)

HUMAN CONTACT

manthan@okoboji.co.in

Prepared for Okoboji Global Private Limited · No. 8, Kranti Kavi Sarvagna Road, Oklipuram, Bangalore 560021 · okoboji.in
This document names credentials by purpose only. It contains no keys, tokens or passwords.

What this page is for

Someone who wants to start or grow a printing business in India has a hard first question: **which machines do I actually need, and can I afford them?** Answering it normally means reading a catalogue, guessing at a workflow, and calling a salesperson to find out that half the list is out of budget.

Print Preneur removes that gap. The visitor says what they want to make in one plain sentence — *“rigid gift boxes in Nashik, ₹4 lakh”* — and a team of agents reads every fact out of that sentence, designs the production line stage by stage from the live Okoboji catalogue, fits it inside the stated budget, flags the bottleneck, and shows what to add later when the budget grows.

Three things it does, all from the same chat box:

- **Design a machine line** for what the visitor wants to produce, at live catalogue prices, inside their budget.
- **Run a feasibility study** for their city — market, rent, break-even and payback, backed by public data.
- **Show where the demand is**, from Okoboji's own record of which machines sell in which cities.

Two rules the system holds to

It never asks a questionnaire — anything the visitor did not say is derived, and every assumption is stated in the reply so a single correction re-plans it. And it never invents a price, a stock status or a product: those come only from the live catalogue.

A human always closes

The page produces a plan, not a contract. The form at the end sends the visitor's query straight to a specialist at **manthan@okoboji.co.in**, who verifies material, size, throughput, installation and the final commercial offer before anything becomes an offer.

The eight agents

Each agent has one job and hands its output to the next. Not every agent runs on every message — the triage agent decides who is needed, which is why a simple price question answers in under a second while a full feasibility study takes longer.

AGENT	JOB	HOW IT DECIDES
1 · Triage	Reads the message and works out what the visitor actually wants.	Deterministic rules first (product, quantity, budget, city are matched by pattern), with the language model only for genuinely ambiguous phrasing. Greetings and machine enquiries are decided by rule so they can never be misrouted.
2 · Catalogue	Finds the real machines and answers price, stock and specification questions.	Searches the live catalogue with the shop's own product copy and SEO data, ranking exact model matches above loose keyword overlap. Every price and stock line comes from the catalogue, never from the model.
3 · Budget ranker	Fits the workflow inside the stated budget.	Treats the budget as a hard ceiling: it orders candidate machines per stage and reports what is funded now versus what needs a later phase. It will not call an unfunded line complete.
4 · Planner	Designs the production line, stage by stage.	A deterministic workflow template per product (print → protect → cut → finish, and so on) filled with real catalogue machines, then explained in plain English.
5 · Web research	Adds independent machine knowledge from the open web.	Searches for the specific model and attaches cited sources. Explicitly barred from supplying price, stock or availability — those stay with the catalogue.
6 · Verifier	Checks the written answer against the facts before it is shown.	Re-reads the draft beside the machine list and corrects any claim the data does not support. This is the last gate before a reply reaches the visitor.
7 · Feasibility	Produces a full economic study for the visitor's city.	Extracts every fact from one message, fills the rest with disclosed assumptions, then builds a 19-section report: market, rent, staffing, break-even, payback and risks, with evidence badges and editable assumptions the visitor can recompute.
8 · Demand map	Answers “where should I start?” from real sales history.	A graph built from Okoboji's own customer records — country → state → city, linked to machine interest — so the answer is where buyers for those machines actually are. Answers in a fraction of a second with no model call at all.

Voice, on top of all eight

The same agents can be reached hands-free. The browser listens, sends each finished sentence, speaks the reply, and reopens the microphone — with the microphone muted while the agents work and while the reply is spoken, so the assistant never transcribes its own voice.

What it runs on

The stack is deliberately small. The web tier has **no runtime npm dependencies at all** — routing, HTTP, JSON and even the outbound mail client are written against the Node standard library. That keeps the supply chain, the image size and the attack surface minimal.

LAYER	TECHNOLOGY	WHY
Web tier	Node.js 22, standard library only	Serves the page and the agent API. No framework, no dependencies to patch.
Front end	Hand-written HTML, CSS and ES modules	No build step and no framework payload — the page is fast on a mid-range Indian phone.
Feasibility service	Python, FastAPI, DuckDB	A separate private service for the economic study and the demand graph, where columnar queries and Python's data tooling pay off.
Storage	DuckDB, plus encrypted fields at rest	Holds the customer graph, cached study responses and captured enquiries. Personal fields are encrypted.
Language model	OpenRouter, free-tier open model	Phrasing, ambiguous-message classification and report prose. Never the source of a price or a fact.
Packaging	Docker Compose, two containers	The public web container and the private feasibility container, on a private network.
Edge	LiteSpeed reverse proxy, HTTPS	Terminates TLS and proxies to the web container. Only this container is reachable from the internet.
Mail	SMTP to the host mail server	Delivers each enquiry to the Okoboji desk. Dependency-free, and the message never leaves the company's own mail server.
Tests	Node test runner and pytest	Over 120 automated tests across both services, covering budget arithmetic, extraction, routing and delivery.

How one message flows

- The visitor's message arrives over HTTPS at the edge proxy and is passed to the web container.
- Triage classifies it and pulls every fact it can from the wording; missing values are assumed and recorded.
- The catalogue is read live from the shop (cached briefly) so prices and stock are current.
- Planner and budget ranker build the line; web research and the feasibility service are consulted only if needed.
- The verifier checks the draft against the machine data, and the reply is returned with its product cards, sources and an agent trace.
- If the visitor submits the form, the enquiry is stored and emailed to a human.

Keys and credentials used

Every credential below is supplied to the running container through environment variables and is never committed to the repository, sent to the browser, or written into a reply. **This document lists purposes only — no key values.**

CREDENTIAL	WHAT IT IS FOR	COST
OpenRouter API key	The only language-model credential in the system. It runs a free-tier open model for phrasing replies, classifying ambiguous messages and writing report prose. Facts, prices and stock never come from it.	Free tier
Store API key and secret	Read-only access to the live okoboji.in product catalogue — names, prices, stock status, images, categories and the shop's own SEO copy.	Owned
Web search key	Independent machine research from the open web, attached to answers as cited sources. Barred from supplying price or availability.	Metered
Maps / places key	Resolves a city to a real location and its commercial context for the feasibility study.	Metered
Internal service token	Authenticates the web container to the private feasibility service, so no one else can call it even from inside the network.	Self-issued
Admin token	Guards the operator console — data imports, graph rebuilds and captured enquiries. Reachable only through an SSH tunnel, never from the public internet.	Self-issued
Encryption key	Encrypts personal fields in captured enquiries at rest.	Self-issued

What the browser can see

Nothing. Every credential stays server-side; the page only ever talks to this service's own API. No key is exposed to page scripts, and the private feasibility service has no public URL at all — it is reachable only through the chat.

What data is connected

Four sources feed the answers. Three are read-only inputs; one is what the page itself collects.

1 · The live Okoboji catalogue

Around **224 live products** read directly from okoboji.in — names, current prices, stock status, images, categories, product URLs, and the shop's own written specifications and feature copy. Cached only briefly, so a price change on the shop shows up in the chat. This is the single source of truth for anything commercial: no price or stock claim is ever generated.

2 · Okoboji's own customer record

A one-time import of the company's sales history — **73,597 rows** consolidated into **40,514 unique customers** after merging duplicates. From it the system builds a demand graph of **2,113 locations** and **4,040 location-to-machine-interest links**, covering **471 paying customers** and roughly **₹5.9 crore** of recorded sales. This is what lets the page answer “where should I start?” with evidence instead of an opinion — for example that Karnataka leads with over 14,000 customers, Bengaluru with over 8,000, and that cutting plotters are the most-wanted machine class.

How that record is protected

The source file lives outside the application entirely — a private directory on the server, excluded from the repository and from the container image, readable only by the system owner. The chat never exposes an individual customer: answers are aggregates, and any group smaller than **20 customers** is suppressed rather than reported.

3 · Public reference data

The Government of India PIN code dataset (data.gov.in) — about **19,000 PIN codes** reduced to a **9,700-place gazetteer** of cities, taluks, districts and states, including old and new name aliases. A visitor can type just a city name and the system derives the state, region and PIN itself. Public market, rent and cost references used in the feasibility study are gathered per study and cited in the report.

4 · What the page collects

DATA	WHY	KEPT
Chat messages in the current session	So the agents remember the brief across turns	In memory, expires after 30 minutes of inactivity
Name, phone, email, company, city, query	Only when the visitor submits the form, so a human can reply	Stored server-side with personal fields encrypted, and emailed to the Okoboji desk
Consent flag	Recorded explicitly — the form cannot be submitted without it	Stored with the enquiry
Coarse request metadata	Rate limiting and abuse protection	The IP address is hashed, never stored in the clear

No advertising or analytics trackers are loaded, and nothing the visitor types is used to train a model.

Where a person takes over

The agents are a planning tool, and the document you are reading is explicit about their limit: **a plan is guidance, not a quotation**. Machine fit, throughput for a specific job, installation, freight, taxes and commercial terms all need a specialist.

The form at the end of the page is that handover. A submitted query — with the plan it refers to, the machines in it and the budget it was fitted to — is delivered by email to **manthan@okoboji.co.in** and stored with a reference number, so nothing depends on a model remembering it. A specialist then verifies every line before it becomes a firm offer.

Reaching a person directly

Email **manthan@okoboji.co.in**, or visit the showroom: No. 8, Kranti Kavi Sarvagna Road, Oklipuram, Bangalore 560021.

Typing “I want a quotation” in the chat also routes straight to the form.

Print Preneur is an AI planning assistant operated by Okoboji Global Private Limited. Machine recommendations, prices and economic estimates are planning guidance based on the live catalogue and public data at the time of the request, and require Okoboji verification before purchase. This document contains no credentials.