# mindgraph AI

Against all odds

CASE STUDIES

# Twenty-two builds, one engineering team.

AI agents, automation layers, and full-stack platforms shipped for manufacturing, logistics, legal, hospitality, agriculture, and enterprise clients.

| 22 | 14 | 17 |
| --- | --- | --- |
| CASE STUDIES | INDUSTRIES | PRODUCTION REPOS |

2026

# Index

Twenty-two engagements, in master-list order.

01 / MANUFACTURING / PAPER & PACKAGING

# Trim-Loss Optimization Engine

A mathematical cutting engine that groups a paper mill’s daily orders into combinations that waste far less paper.

Client: **Gujarat-based paper manufacturer**

Manufacturing | Paper & Packaging | Linear Programming | Cutting Optimization | Dashboard


## PROBLEM

A kraft paper manufacturer running roughly 250 tons of paper a day was cutting bulk jumbo rolls into the many smaller roll sizes clients actually order — sorted by color, thickness, sizing, price, and minimum order quantity. Planning those cuts by hand left an average of 5% of every roll as trim waste, and at close to $1,000 a roll, that waste was costing the business real money every single day.

## OUR SOLUTION

**A mathematical cutting engine that groups a paper mill’s daily orders into combinations that waste far less paper.** Every morning’s orders are automatically grouped by what’s available and what clients actually asked for, then matched into cutting combinations that pair similar and complementary orders together instead of cutting each one in isolation. The result showed up directly in the numbers: average trim loss came down from around 5% to somewhere between 1.5% and 2%, on a business cutting hundreds of tons of paper a day.

## TECHNICAL DETAILS

### Cutting Optimization Engine

A linear-programming model takes the day’s incoming orders — grouped by color, thickness, sizing, price band, and minimum order quantity — and searches for combinations of orders that can be cut from the same jumbo rolls with the least leftover trim.

### Daily Planning Dashboard

A dashboard surfaces each day’s grouped cutting plan for the floor team, replacing manual, experience-based grouping with a plan generated fresh every morning from that day’s actual order book and available stock.

### Stack

Python for the optimization engine, Node.js for the application layer.

02 / MANUFACTURING / SALES AUTOMATION

# Microsoft 365 Inbox to ERP-Validated Quote Generator

An AI agent that reads incoming material requests, checks them against the ERP, and drafts the quotation before a person opens the email.

Client: **Mumbai-based tools manufacturing conglomerate**

Manufacturing | Email Automation | ERP Validation | Document Generation | OCR

PIPELINE — GROUPED STEPS

| IN | INGEST | PARSE | CLASSIFY | OUT |
|---|---|---|---|---|
| **Graph Fetch**<br>*pull the message via Microsoft Graph API* | **Extract Body**<br>*HTML → text, strip banner*<br>**Route Attachment**<br>*by content-type* | **Image OCR**<br>*AI OCR per attachment*<br>**Doc Parse**<br>*XLS→XLSX, Docling*<br>**PDF Render**<br>*PyMuPDF @300 DPI, OCR'd* | **Merge Document**<br>*body + attachments → one text*<br>**LLM Classify**<br>*standard / mixed / non-standard* | **ERP Validate & Quote**<br>*per-location split, auto-drafted reply* |

## CONTEXT

Customer material requests arrived by email — sometimes typed, sometimes a photographed handwritten list, sometimes a scanned PDF or spreadsheet — and every one required someone to read it, manually check whether each material existed in the ERP, work out which of the company's locations could fulfil it, and type up a quotation by hand. A single multi-item order could take a sales coordinator the better part of a day, and mixed orders — some items valid, some not — had no clean handling path.

## OUR SOLUTION

An AI agent that reads, checks, and drafts. It merges the email body with OCR'd images/PDFs and parsed Office attachments into one document, classifies the inquiry, and extracts every material number requested. Each material is validated live against the ERP; inquiries end up correctly labelled standard, mixed, or non-standard based on what's actually fulfillable — not just what was asked for. Valid materials are grouped by fulfilment location and turned into one quotation document per location, attached to an auto-drafted reply.

## TECHNICAL DETAILS

- Multi-format ingestion: AI OCR for images and scanned PDFs (300 DPI/page), document parsing for Word/PPT/XLS
- LLM classification (standard / mixed / non-standard) with automatic chunking and result-merging for long threads
- Live ERP validation per material line over a secured database connection
- Automatic reclassification when an inquiry contains both valid and invalid materials
- Location resolution mapping native material codes to fulfilment-location codes
- Multi-location quotation splitting — one document generated per fulfilment location
- Thread-aware processing: skips stale messages, cleans up duplicate drafts, auto-labels processed mail

03 / LOGISTICS / LEGACY SYSTEM AUTOMATION

# No-API Legacy ERP-to-REST Bridge

Turning a click-only ERP portal into three REST endpoints that return a finished Excel report.

Client: **Maharashtra-based transportation firm**

Logistics | Legacy System Integration | Browser Automation | Report Generation

PIPELINE — VERTICAL FLOW

- IN **REST Request** *GET /report/:id + date range*
- 01 **Pre-Clear Session** *hit Logout.aspx before every run*
- 02 **Login** *auto-recovers from "already logged in"*
- 03 **Select FY** *derives FY from date, nested iframe*
- 04 **Trigger AjaxPro Search** *types report ID, intercepts raw POST*
- 05 **Parse & Click Result** *regex the <LI> list, click by DOM value*
- 06 **Fill Filters** *Category / Class / Group / Date Type*
- 07 **Show Report** *settle wait, new-tab-or-same-page*
- 08 **Export → Excel** *cross-frame export icon*
- OUT **Intercept & Logout** *context-level download catch*

## CONTEXT

The firm's core operations ran on a legacy ASP.NET ERP with no API — every report meant a person logging in, navigating nested iframes, searching for the report by name, setting filters by hand, and exporting through a Crystal Reports viewer. The portal had no tolerance for concurrent sessions — a second login without logging out first would lock the account.

## OUR SOLUTION

An automation layer that drives the portal itself and exposes it as three simple REST endpoints: login (with recovery from "already logged in" conflicts), the correct financial year selected, and the right report found by reading the portal's own search response rather than guessing at screen positions. It fills in the filters, triggers the export, and returns a ready Excel file — session always closed cleanly, success or failure.

## TECHNICAL DETAILS

- Playwright automation over a legacy ASP.NET portal with no public API, resilient to layout changes
- Session-conflict handling: automatic "Force/Continue" detection and re-login
- Report resolution via AjaxPro response interception, not visual/coordinate matching
- Crystal Reports export automation with multi-frame export-button discovery
- Context-level download interception, catching downloads from nested sub-frames
- Guaranteed logout in a finally block every run; FastAPI wraps each report as a REST endpoint

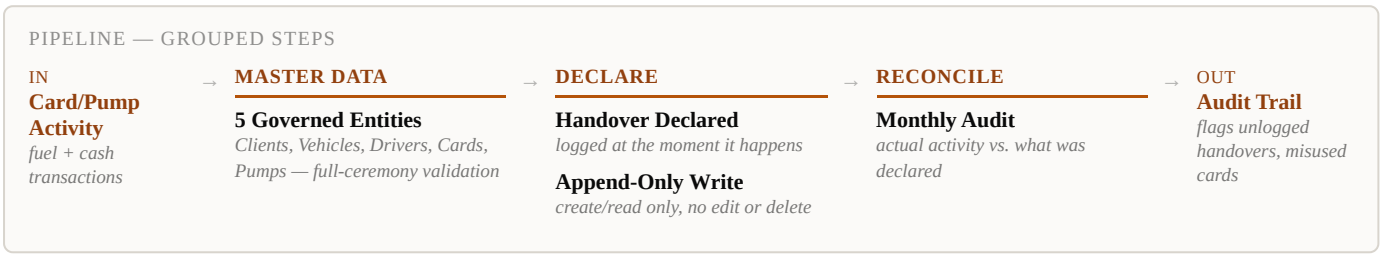
04 / LOGISTICS / FLEET DATA GOVERNANCE

# Fuel Expense Reconciliation Ecosystem

Replacing informal card handoffs and untracked vehicle assignments with a system the team is required to keep honest.

Client: **Maharashtra-based transportation firm**

Logistics | Fleet Management | Data Governance | Audit Trail | FastAPI




## CONTEXT

The client runs trucks against fuel and cash sources — oil-company cards, credit at partner pumps, cash-advance cards — reconciled entirely by hand in Excel by a team of three. Underneath the reconciliation problem sat a governance problem: which card was on which vehicle, and who was driving it, drifted constantly and informally, with no system requiring anyone to declare a handover when it happened — so nobody could say with confidence whether a card firing for a given vehicle was legitimate or a policy violation waiting to be discovered at month-end.

## OUR SOLUTION

A governed master-data system built specifically to stop tolerating that drift. Every vehicle, driver, card, and pump lives in a full-ceremony record — no inline shortcuts, no partial entries accepted where the business rules say a field is required. Sitting on top of that master data is an append-only assignment log: every card handover and driver reassignment has to be declared at the moment it happens, and the record can never be edited or deleted afterward. That gives the client a trustworthy monthly audit trail, flagging exactly the handovers that went unlogged or the cards that turn up somewhere they were never assigned.

## TECHNICAL DETAILS

### Master Data

Five governed entities — Clients, Vehicles, Drivers, Cards, Pumps — each enforced through full-ceremony forms and matching API-level validation. Ownership status has no default and must be explicit; a normalizer converts legacy number-first registration formats into standard plate order.

### Compliance Ledger

The Card/Driver assignment log is append-only by design — only creation and read endpoints exist, with no update or delete path, so the audit trail itself can’t be altered after the fact.

### Stack

FastAPI + asyncpg over Postgres, raw SQL with the schema file as the DDL source of truth, and a React/TypeScript frontend covering every master-data page plus assignment declaration and history.

05 / LEGAL SERVICES / MEETING INTELLIGENCE

# Secure Real-Time Multilingual Transcription Engine

A transcription engine that joins the call, translates as it listens, and never loses a word even when the connection drops.

Client: **Premier legal firm, UK & Dubai**

Legal Services | Real-Time Transcription | Multilingual | LLM Correction | Encrypted Data

PIPELINE — VERTICAL FLOW, GLOSSARY FEEDS DIRECTLY INTO CORRECTION

- IN **Meeting Bot Joins** *Teams / Zoom, live audio*
- 01 **Live Transcribe** *language ID + live translation to English*
- 02 **Glossary Build** *pre-call context parsed into structured terms*
- 03 **LLM Correction** *checked against glossary, spacing untouched*
- 04 **Auto-Reconnect** *rolling buffer replays audio on a drop*
- 05 **Post-Call Re-Transcribe** *full audio, diarized, multi-segment merge*
- 06 **Mixed-Language Preserved** *never auto-translated into one language*
- OUT **Encrypted Transcript** *encrypted Postgres, API-key auth*

## PROBLEM

Cross-border legal meetings mix languages and highly specific terminology — client names, matter names, technical and legal terms — that generic transcription tools consistently mangle. There was no reliable way to get a clean, unified English transcript in real time, and recordings needed to be handled as sensitive data throughout, not left in a plaintext database.

## THE CONTRIBUTION

A purpose-built transcription system that joins the meeting, transcribes and translates live, and corrects itself against that meeting's own vocabulary as it goes. Background the team provides is parsed into a structured glossary beforehand; every line is then checked against it by an LLM correction pass that fixes names and terms without touching spacing or wording. If the connection drops, a rolling buffer replays audio so the transcript never gaps, and a post-call pass produces a cleaner, diarized final transcript.

## TECHNICAL DETAILS

- Real-time pipeline: meeting bot → speech engine (language ID + live translation) → LLM correction → display
- Per-meeting glossary generation, fed into both live and post-meeting correction passes
- Correction engine preserves exact token spacing and streaming partial text — no autocomplete, no rephrasing
- Automatic reconnect on speech-engine session limits, with a rolling buffer replaying recent audio
- Post-meeting pass: full re-transcription, multi-segment diarization and merge
- Encrypted Postgres database, key derived from an admin password, never persisted in plaintext
- API-key authentication with role-based (admin/user) access control

06 / SOFTWARE · SAAS / SALES ENABLEMENT

# Real-Time Sales Call Assistant

An invisible second brain on every sales call, answering from the company’s own product material before the rep even finishes hearing the question.

Client: **Bangalore-based product-first SaaS startup** B2B SaaS | Sales Enablement | Real-Time RAG | Live Call Assistant


## CONTEXT

Reps selling a fast-moving product were expected to have instant, accurate answers to detailed spec, pricing, and integration questions — the kind of thing that lives scattered across product sheets, pricing docs, and internal battlecards, not in anyone’s memory. A wrong or hedged answer mid-call cost credibility; digging for the right document while a prospect waited cost momentum.

## THE CONTRIBUTION

Built a meeting bot that joins the call and works entirely on the rep’s side of the conversation. Before the call, it’s primed with a RAG index built from the company’s own product sheets, pricing material, and battlecards. During the call, it listens live, detects when the client has asked something answerable from that material, retrieves the relevant passages, and surfaces a short, source-grounded suggestion through a hoverable overlay visible only to the rep — invisible to the client on the other end of the call. The rep stays in control of the conversation; the copilot just makes sure the right fact is one glance away instead of a scramble.

## TECHNICAL DETAILS

- Meeting bot joins the call via the video platform’s own bot/API layer, listening on the rep’s session
- Pre-call RAG ingestion: product sheets, pricing docs, case studies, and battlecards chunked and embedded per team
- Streaming speech-to-text plus a lightweight question-detection layer identifying retrievable utterances in real time
- RAG retrieval against a team-scoped Postgres/pgvector store, ranked by relevance and recency
- LLM-composed, source-grounded suggestions sized for a glance, not a document dump
- Overlay rendered client-side only, excluded from screen-share capture
- Sub-2-second target latency from spoken question to surfaced suggestion
- Encrypted Postgres storage with team-scoped document isolation and API-key authentication

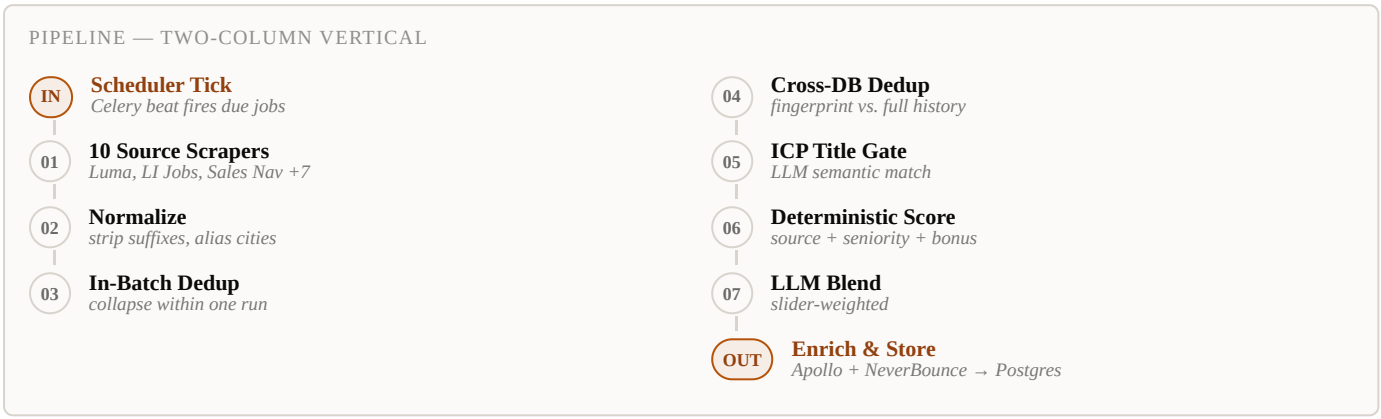
07 / SOFTWARE · SAAS / GTM INTELLIGENCE

# Multi-Source Lead Scraping & Scoring Engine

An 11-platform prospecting habit, replaced by one scored pipeline.

Client: **B2B ESG SaaS company (scaling outbound)**

B2B SaaS | Lead Generation | Multi-Source Scraping | LLM Scoring | Deduplication




## PROBLEM

Manual prospecting across 11+ platforms — job boards, events, LinkedIn, news, regulatory filings — with no dedup between them and no shared scoring logic. Reps re-discovered the same prospects under different names, and pipeline throughput was bottlenecked by manual research time, not lead availability.

## WHAT SHIPS NOW

**10 scraper sources → 1 table.** Luma, LinkedIn Jobs, LinkedIn New Hires, Sales Navigator, ICP News, Competitor News, Blog, LinkedIn Commenters, ESG Regulatory Signals, and Slack-native inbound — all on independent schedules, landing in one deduplicated Postgres table.

**4-key dedup, every lead, every time.** LinkedIn slug, email, phone, and name+company+city fingerprinting — checked in-batch and against the full historical database before a single enrichment credit is spent.

**99 titles, 9 domains, one filter.** Every profile is scored against the client's own ICP title map before it's allowed anywhere near the scoring engine.

**Transparent scoring, not a black box.** A deterministic point system blended against an LLM score on a slider the client controls.

## TECHNICAL DETAILS

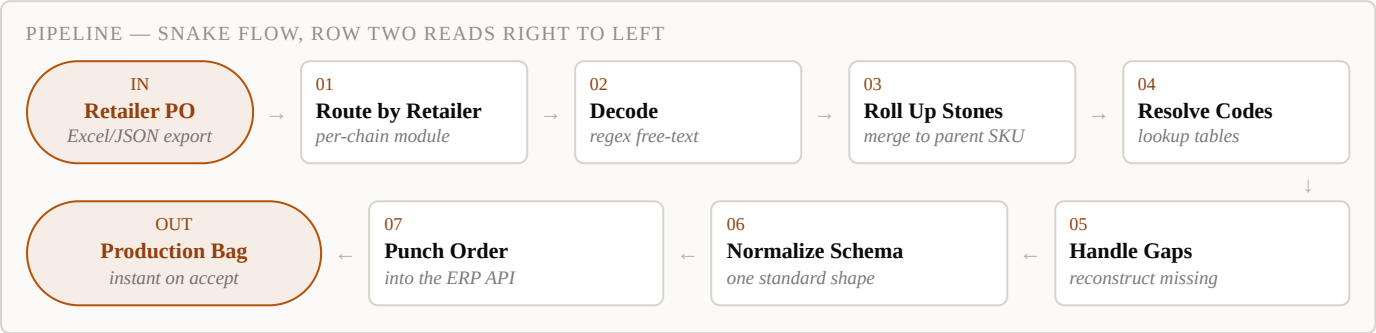
- 10 independent scraper sources unified into one Postgres lead table via Celery beat scheduling
- 4-key deduplication fingerprinting, in-batch and against full DB history
- ICP relevance filtering against a 99-title map across 9 ESG domains, LLM-evaluated per profile
- Deterministic + LLM blended scoring engine, transparent point breakdown, adjustable via a slider
- Company compliance-framework resolution (BRSR, CSRD, GRI, TCFD) cached per account
- On-demand Apollo contact enrichment and NeverBounce email validation before handoff to sales
- Slack-native inbound signal capture (website-visitor identification → structured lead)

# Multi-Retailer PO-to-Production ERP RPA Layer

A central engine that normalizes every retailer’s purchase-order format into one standard schema before it ever touches production.

Client: **Jewellery manufacturer ($180M revenue)**

Manufacturing | Jewellery | RPA | ERP Integration | Order Automation




## PROBLEM

Every major retail chain the manufacturer supplied sent purchase orders in its own idiosyncratic format: attributes like size, finding, and stone-setting encoded inside free-text descriptions rather than their own columns, diamond weight and piece data split across separate line-items needing manual roll-up to the right SKU, and retailer-specific code tables for tone, stamping, and certification that had to be looked up by hand. Getting an order correctly into production meant a person reading, decoding, and re-typing every one of those idiosyncrasies — per retailer, per order.

## OUR SOLUTION

A central engine that normalizes every retailer’s order format into one standard schema before it ever touches production. A dedicated transformation module per retail chain decodes that chain’s encoded descriptions, rolls up nested diamond weight and piece counts onto the correct SKU, and resolves tone/stamping/certification codes automatically via lookup tables — with documented rules for reconstructing missing attribute data rather than kicking every irregularity to a human. Once normalized, the order is punched directly into the manufacturer’s system, and a production bag is generated the instant the order is accepted.

## TECHNICAL DETAILS

- Per-retail-chain transformation modules, each mapping that chain’s raw export into one standard schema
- Regex-based extraction of encoded product attributes from free-text descriptions
- Diamond weight and piece-count roll-up: nested stone line-items merged onto their parent SKU row
- Lookup-table-driven resolution of tone, stamping, and certification codes per retailer
- Documented exception-handling rules reconstructing missing attribute segments, not failing the row
- Automatic client-code, book-name, and auth-header routing into the order-import API
- Automatic production-bag generation triggered immediately on a successful order punch

09 / FINANCIAL SERVICES / IDENTITY VERIFICATION

# Real-Time KYC Verification API

A document-quality gate and an AI reader working together, so a bad photo never wastes an extraction call and a good one gets structured data back in seconds.

Client: **Banking & financial services firm**

Banking | KYC | Multilingual OCR | Computer Vision | Async Processing

PIPELINE — GROUPED STEPS

IN
**Image Submit**
*task ID returned immediately, HTTP 202*

→

**QUALITY GATE**

**Blur**
*Laplacian variance*

**Resolution**
*minimum pixel threshold*

**Contrast, Glare, Darkness**
*rejection checks*

**Edge Density**
*catches near-blank frames*

→

**PREPARE**

**Contour + Deskew**
*four-point outline, perspective transform*

→

**EXTRACT**

**VLM Extraction**
*strict JSON-only response contract*

→

OUT
**Poll for Result**
*structured JSON, or a specific rejection reason*

## CONTEXT

Verifying identity and banking documents at volume meant handling whatever a customer’s phone camera actually produced — blurry, glared, dark, skewed, or genuinely unreadable images mixed in with good ones — across documents issued in multiple languages and scripts, for a use case where extraction accuracy on sensitive financial and identity data wasn’t negotiable. Sending every image straight to a vision model regardless of quality wasted calls on documents that were never going to extract cleanly, and a synchronous endpoint would leave the API blocked under real load.

## THE CONTRIBUTION

Built an asynchronous KYC document API that filters before it extracts. Every submitted image passes through a six-stage quality gate — blur, resolution, contrast, glare, darkness, and edge density — before anything is sent to an AI model at all; a document that fails any check is rejected immediately with a specific, actionable reason. Images that pass are automatically detected by their four-point outline and corrected with a perspective transform to a clean top-down view, then sent to a self-hosted, open-source vision-language model — chosen for high accuracy across multiple languages and document types — under a strict JSON-only extraction prompt. The whole pipeline runs asynchronously on a Celery worker pool backed by Redis, with automatic retries and guaranteed cleanup of every image file regardless of outcome.

## TECHNICAL DETAILS

- Six-stage pre-AI image quality gate: blur (Laplacian variance), resolution, contrast, glare, darkness, edge density
- Four-point document contour detection with perspective-transform deskewing before extraction
- Extraction via a self-hosted, open-source vision-language model under a strict JSON-only contract
- Fully asynchronous FastAPI + Celery + Redis pipeline: submit → immediate task ID → poll for result
- Automatic retries (up to 3) with backoff on transient task failures
- Hard/soft task time limits (5 min / 4 min); results expire after 30 minutes
- Guaranteed image file cleanup after every run, success or failure
- API-key authenticated endpoints

10 / SPORTS & FACILITIES / UNIFIED ERP + DIGITAL PLATFORM

# MultiFacility Sports Ecosystem ERP

A single ERP and digital platform spanning an entire sports city complex — bookings, membership, academy, and facilities on one shared data layer.

Client: **Large multi-facility sports city complex, Dhaka, Bangladesh**

Sports & Facilities | ERP | AI Chatbot | Mobile & Web | POS

PIPELINE — VERTICAL FLOW, THE MEMBER'S OWN JOURNEY

- IN **Member App / Website** *booking, enrollment, ticketing entry point*
- 01 **Facility Availability** *real-time booking across every facility*
- 02 **Chatbot / WhatsApp Support** *instant help with bookings, confirmations*
- 03 **Payment** *cards, wallets, local rails, auto reconciliation*
- 04 **QR/NFC Card Issue** *digital card tied to gate pass and POS*
- 05 **Access & POS Scan** *gate pass, facility access, point of sale*
- 06 **Shared Data Sync** *academy, inventory, HR, financial modules*
- OUT **Live Dashboards** *real-time facility, revenue, academy analytics*

## PROBLEM

A sports complex running a dozen facilities, a training academy, residential housing, retail, and hospitality partnerships had no single operating system tying any of it together. Bookings, membership records, maintenance logs, and academy enrollment each lived in their own process — often on paper — with no shared view of occupancy, revenue, or equipment health, and no self-service option for members to book or pay for anything themselves.

## OUR SOLUTION

A single ERP and digital platform spanning the entire complex: membership with digital QR/NFC cards, real-time booking across every facility, academy management with coach and player tracking, plus residential, inventory, HR, and financial modules on one shared data layer. Members get a website and mobile app for booking, enrollment, event ticketing, and a personalized dashboard — backed by an AI chatbot and WhatsApp bot for instant support. Live dashboards replace month-end reporting with real-time facility, revenue, and academy analytics.

## TECHNICAL DETAILS

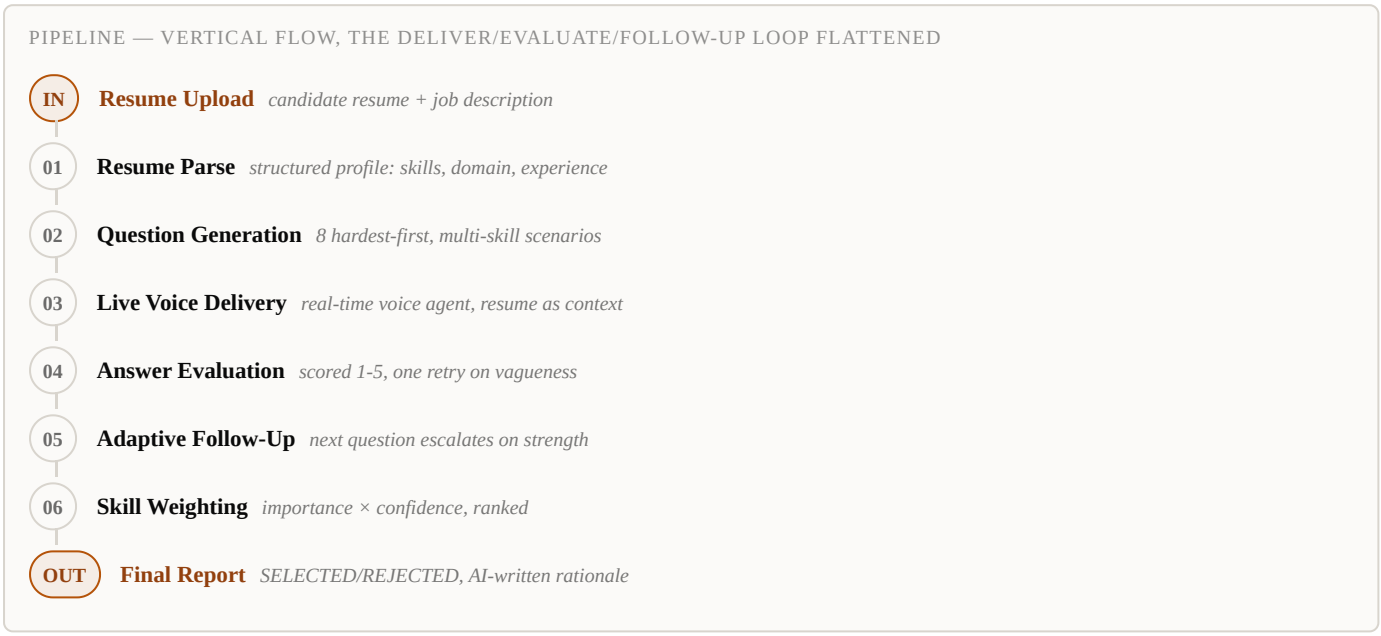
- Microservices architecture: Node.js/Express backend, React.js web frontend, React Native mobile app
- PostgreSQL for relational data, MongoDB for document storage, Supabase for real-time data sync
- JWT authentication, TLS 1.3 encryption, GDPR-aligned data handling
- AI chatbot and WhatsApp bot for member inquiries, booking status, and confirmations
- Digital membership cards (QR/NFC) tied to gate pass, facility access, and POS at point of sale
- Payment gateway integration (cards, wallets, local payment rails) with automated reconciliation
- KPI dashboards across facility usage, revenue, and academy performance, with custom reporting
- Built and load-tested for concurrent access at scale across a multi-facility campus

# Elite-Standard Screening, Automated

A two-layer AI interviewer that builds candidate-specific questions and holds every answer to the same elite, consistent bar.

Client: **Enterprise recruiting clients**

HR Tech | Recruiting | Voice AI | Adaptive Interviewing | Automated Scoring




## PROBLEM

Screening candidates at volume meant choosing between generic scripted questions that never tested what was actually on a resume, or spending real recruiter time per candidate to judge consistently. Vague or rehearsed answers routinely got through — nobody had a consistent enough standard to push back.

## THE CONTRIBUTION

Built a two-layer AI interviewer. The candidate’s resume is parsed into a structured profile and matched against the job description to generate hardest-first, multi-skill scenario questions specific to that candidate, delivered as a live voice conversation with the resume fed in as context. Vague or evasive answers get exactly one retry before being scored a fail, and follow-up questions escalate in difficulty based on the candidate’s own previous answer.

## THE REPORT IS A DECISION, NOT JUST NOTES

Skills are weighted by importance and confidence into ranked strengths and weaknesses, normalized against a configurable pass threshold, for a clear SELECTED/REJECTED recommendation with AI-written rationale — the same bar, every time.

## TECHNICAL DETAILS

### Resume-to-Questions

LLM-based resume parsing produces a structured candidate profile, matched against the job description to generate 8 hardest-first, multi-skill scenario questions unique to that candidate.

### Live Voice Delivery

The interview runs as a real-time voice conversation via an AI voice agent, with the parsed resume passed in as live context so questions and follow-ups reference it naturally.

### Adaptive Difficulty

Follow-up questions are generated dynamically based on the candidate’s previous answer, escalating difficulty and integrating multiple skills rather than repeating a fixed script.

### Scoring & Reporting

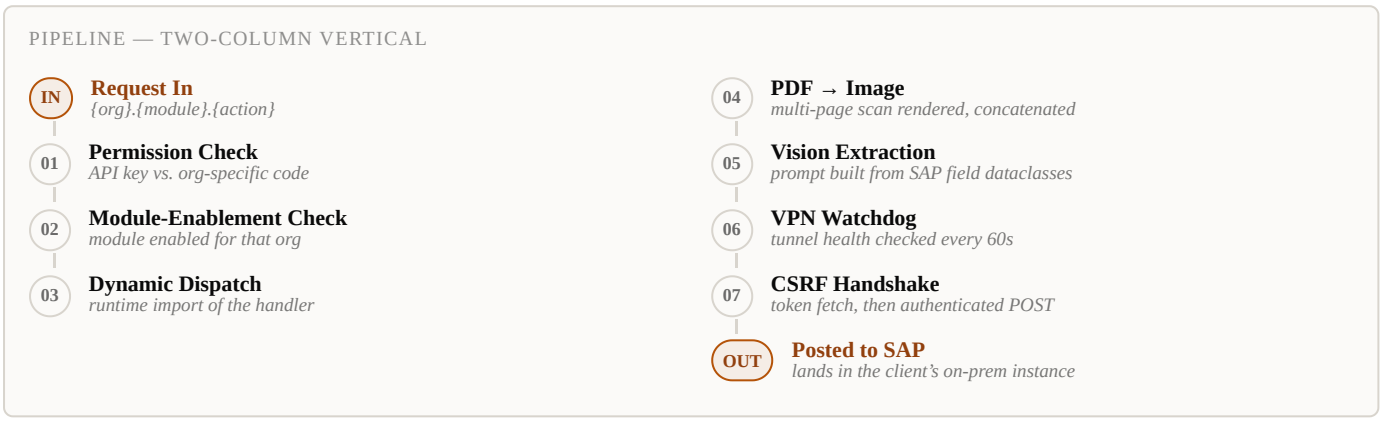
Every answer is scored 1-5 with vagueness detection (one retry, then a fail) and per-skill confidence ratings, weighted by importance × confidence into ranked strengths and weaknesses for a SELECTED/REJECTED recommendation. Fallback logic ensures a usable report even if AI commentary generation fails.

# SAP RPA Platform

Multi-tenant platform that reads scanned SAP documents with AI and posts them straight into SAP over a self-healing VPN tunnel.

Client: **Manufacturing company running SAP ERP**

Manufacturing | SAP | RPA | Document Automation | Multi-Tenant




## PROBLEM

Purchase requisitions, sales quotes, and invoices arrived as scanned PDFs, and someone had to read each one and key it into SAP by hand, field by field. SAP's own APIs sit behind a CSRF-token handshake and, for this client, a VPN tunnel to an on-prem instance — not something a simple integration script could just point at.

## OUR SOLUTION

A multi-tenant RPA platform that reads documents with AI and posts them straight into SAP. Each organization's integration logic is resolved dynamically at request time, so a new client is a configuration change rather than a rebuild. Incoming PDFs are read by GPT-4o Vision using an extraction prompt generated directly from SAP's own field definitions, and a dedicated watchdog service keeps the client's VPN tunnel to their on-prem SAP instance alive around the clock.

## TECHNICAL DETAILS

### Dynamic Module Router

A single API endpoint resolves {org}.{module}.{action} to a runtime-imported, organization-specific handler, checking a permission string and per-org module-enablement flag before dispatch.

### Document Extraction

Multi-page PDFs are rendered to images, concatenated into one, and read by GPT-4o Vision. The extraction prompt is generated directly from the SAP field dataclasses, so the JSON returned always matches SAP's exact schema.

### SAP OData Integration

Posting to SAP replicates its CSRF-protected OData handshake exactly: a token fetch, then an authenticated POST in the same session, matching SAP's required headers field for field.

### VPN Watchdog

A standalone FastAPI service manages the OpenFortiVPN tunnel, with a 60-second self-healing monitor that reconnects automatically unless manually turned off.

### Multi-Tenant Administration

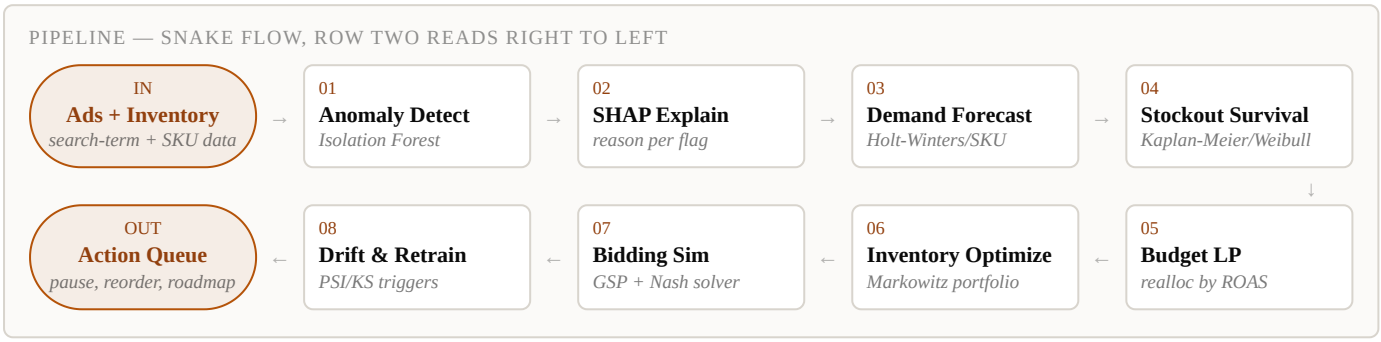
Organizations, users, API keys, module enablement, and per-org credentials sit on a fine-grained permission model, managed through role-scoped Super Admin and tenant Admin dashboards.

# PPC & Inventory Intelligence Platform

A platform that watches every dollar of Amazon ad spend and every unit of inventory, and tells you exactly what to do about both.

Client: **Amazon-based exporter**

E-Commerce | Amazon PPC | Demand Forecasting | Optimization | MLOps




## PROBLEM

- $1M/month GMV, scaling toward $2M+ — inefficiencies were set to scale right along with it
- 52.5% of SKU-days sat at zero available inventory
- 29% of PPC spend wasted — zero-sale campaigns, runaway ACOS up to 267% on some
- 6% repeat customer rate against a 20–25% category benchmark
- 11–12% return rate, worst in the best-selling category
- Launches and budget calls made on gut feel — no demand signal behind either

## OUR CONTRIBUTION

- PPC engine auto-pauses zero-sale campaigns and flags runaway ACOS before it burns budget
- Demand forecasting predicts stockouts 90 days out and triggers reorders against real lead times
- Inventory depletion modeled with survival analysis, not a flat reorder point
- Return-risk scoring flags high-risk orders at the moment of purchase
- Every model watched for drift and retrained automatically once its inputs move

## RESULTS

- PPC waste cut 50–60% within 90 days
- SKU availability up from 47% to 70%+
- Repeat purchase rate roughly doubled
- $1.4M–$2.1M/year in recoverable value identified, sequenced on an ICE-scored roadmap
- Net contribution margin improved from ~26% to ~33.5%

## TECHNICAL DETAILS

### Anomaly Detection & Explainability

- Isolation Forest over search-term-level spend, clicks, impressions
- Every flag paired with a SHAP explanation — not just a score

### Demand Forecasting & Inventory Survival

- Holt-Winters per SKU, weekly seasonality, backtested confidence bounds
- Kaplan-Meier and Weibull hazard models estimate time-to-stockout per SKU

### Optimization

- LP solvers reallocate PPC budget across campaigns by expected ROAS
- Markowitz-style portfolio optimization, carrying-cost minimization

### Competitive Bidding Simulation

- GSP auction simulator, Nash-equilibrium bidding solver, multi-round bid-war simulator
- Tests a bidding strategy against modeled competitors before real budget commits

### Drift Monitoring & Retraining

- PSI and Kolmogorov-Smirnov tests catch feature drift against each model’s baseline
- A drifted model is flagged and routed through automatic retraining

# Personal Injury Case Management Platform

An MVP case management platform connecting the law firm, the patient, and every provider involved in a case, with the access controls a legal and medical record actually needs.

Client: **Personal injury law firm**

Legal Tech | Personal Injury | MVP | Compliance | Case-Scoped AI

PIPELINE — GROUPED STEPS

| IN | ACCESS | ACT | GOVERN | OUT |
|---|---|---|---|---|
| **Case Request Created**<br>*firm opens a case needing patient/provider action* | **Magic Link Issued**<br>*single-use, expiring, hashed-at-rest*<br>**Scoped Portal Opens**<br>*session-less, scoped to just this case* | **Action Taken**<br>*consent, upload, message / fulfill, confirm slot*<br>**Preflight Gate Check**<br>*per-provider-type requirements checked off* | **Audit Log Write**<br>*every sensitive read/write, append-only*<br>**Role-Scoped Visibility**<br>*firm roles see only their own scope* | **Case-Scoped AI**<br>*answers only from docs the user could already see* |

## PROBLEM

A personal injury case touches a law firm, the patient, and a rotating cast of clinics and providers — each on their own phone, email, and paper trail, with sensitive medical and legal information passed around with no consistent record of who saw what or consented to what. A firm juggling dozens of active cases had no single system connecting case status, provider requests, billing, and patient consent.

## OUR SOLUTION

**An MVP case management platform connecting the law firm, the patient, and every provider involved in a case, with the access controls a legal and medical record actually needs.** Patients check in, sign consent forms, upload documents, and message the firm through a link that needs no account or password; providers get their own scoped portal to fulfill requests, confirm appointment slots, and see only the billing information relevant to their own involvement. Every role on the firm side sees exactly the case data their job requires, with every sensitive read and write logged to an audit trail nobody, not even an admin, can edit after the fact.

## TECHNICAL DETAILS

### Magic-Link Patient & Provider Portals

A single-use, expiring, hashed-at-rest magic link opens a session-less portal scoped to just their case. Patients check in, e-sign and revoke consent, upload documents, message the firm, with every action recorded in a receipt ledger.

### Role-Based Access With a Read-Only Axis

Four firm-side roles each see only the case data their job requires, with per-org permission overrides. A separate read/write flag applies across any role, so a broad read-only reviewer falls out of the same mechanism.

### Compliance-Grade Audit Logging

Every write, and every sensitive read, is logged automatically by the request middleware. The audit log is append-only at the database level: an update is rejected outright, a delete only permitted from the scheduled retention job.

### Provider Fulfillment With Preflight Gates

A request sent to a clinic or specialist can’t be marked fulfilled until every requirement specific to that provider type is checked off, so a completed request always means the same thing.

### Case-Scoped AI Assistant

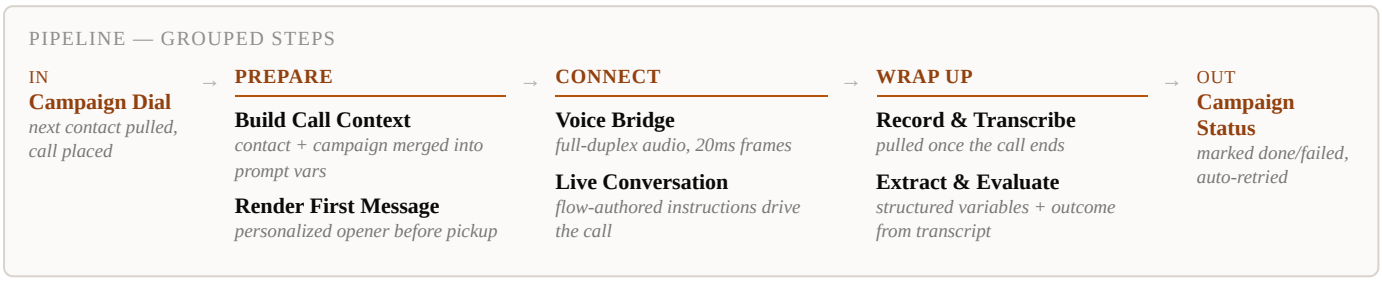
Answers questions grounded only in that case’s own documents, mirrored under the same role-based visibility rules the rest of the app enforces — it never answers from a document the asking user couldn’t otherwise open.

# AI Voice Calling Platform

Multi-tenant platform that runs AI voice agents over real phone lines — outbound campaigns and inbound reception, both handled by the same conversational AI.

Client: **Enterprise pilots across fashion retail, distribution, and steel manufacturing**

Voice AI | Telephony | Conversational AI | Multi-Tenant | Compliance




## PROBLEM

Outbound calling and inbound reception ran on separate systems, neither able to scale past a human call center's capacity. Every enterprise client needed its own phone numbers, its own KYC-verified telephony account, and its own conversation logic — without waiting on a shared platform team to configure it.

## OUR SOLUTION

**A multi-tenant platform where AI voice agents run outbound campaigns and answer inbound calls over real phone lines.** Conversational AI is bridged directly into carrier-grade telephony in real time, and every organization gets its own compliance-verified telephony account, self-serve and fully isolated from every other tenant. Conversation logic is written once as a structured flow, validated by AI, and reused across every call the agent takes.

## TECHNICAL DETAILS

### Real-Time Voice Bridge

A WebSocket bridge relays full-duplex audio between the telephony carrier and the conversational AI engine live, resampling audio between each system's native format in 20ms frames so turn-taking stays natural.

### Multi-Tenant Telephony

Each organization gets its own carrier sub-account, funded independently, provisioned automatically the first time it completes KYC. Number search and purchase is fully self-serve, with every tenant's numbers and calls isolated.

### KYC Compliance Enforcement

Verification happens on the carrier's own hosted widget. A background task enforces a 7-day compliance deadline, backed by a reconciler that independently re-checks in-flight verifications — status is always re-verified server-side.

### AI-Validated Conversation Flows

Conversation logic is authored as a structured flow and converted into the agent's operating instructions by an LLM, with a second AI validation pass checking it before it's allowed to go live.

### Campaign Execution & Custom Integrations

Outbound campaigns run as background jobs that dial down a contact list and retry failed connections automatically, with client-specific CRM integrations built per engagement on the same core platform.

# Field-to-Dispatch Operations Platform

A role-based mobile ERP that replaces free-text WhatsApp logging across a banana export operation’s entire field-to-dispatch chain.

Client: **Banana export operation, Western India**

Agriculture | Export Operations | Mobile ERP | Role-Based Access | Dashboards

PIPELINE — TWO-COLUMN VERTICAL

- IN — **Raw Harvesting Export** — *daily field log, xlsx*
- 01 — **Validate Headers** — *template mismatch caught early*
- 02 — **Resolve Location** — *village → location mapping table*
- 03 — **Apply Location Rules** — *region-specific formula exceptions*
- 04 — **Determine Packing Type** — *per row*
- 05 — **Build Per-Date Rows** — *recovery %, box totals, volume conv.*
- 06 — **Flag for Review** — *unmapped/ambiguous → human*
- OUT — **Dashboard Workbook** — *per-date production dashboard*

## PROBLEM

Every stage of the harvest-to-dispatch pipeline — field visits, rate negotiations, box counts, dispatch records — was logged as free-text WhatsApp messages, formatted differently depending on who typed it. Nothing was searchable, rates couldn’t be audited after the fact, and dispatch counts had to be manually reconciled against paper transport notes.

## OUR SOLUTION

**An ERP covering the full field-to-dispatch chain, from vendor mapping through cold storage arrival, as one structured system.** Fourteen operational steps that used to be free-text messages are now mobile forms with mandatory fields, camera-only capture, and automatic GPS and timestamp embedding. The system computes recovery percentages, box totals, and volume conversions on its own, surfacing results across ten live dashboards — starting with an automated daily production dashboard reproducing the summary report regional staff used to build by hand every day.

## TECHNICAL DETAILS

### Structured Field Capture

Every one of the fourteen workflow steps is a mobile form with mandatory fields, not free text. Photo/video capture is camera-only, and every capture embeds GPS location and a timestamp automatically.

### Automatic Derivation & Hidden Business Rules

Recovery percentages, box totals, and volume conversions are computed automatically. A location-rules table encodes exceptions confirmed from real records — including one region whose deduction formula is always zeroed. Anything ambiguous is flagged for a human, never guessed.

### Role-Based Access Control

Nine distinct roles — from field scout through rate negotiator to cold storage QC — each see only the steps relevant to them, with attribute-based access since one employee can carry more than one role.

### Operational Dashboards

Ten live dashboards cover everything from an executive daily overview to rate trends, recovery % by vendor and hub, C-Class rejection rates, and a full audit/change log of every record edit.

### Built for the Field

Python backend, Node.js/TypeScript application layer, PostgreSQL, Celery/Redis for async processing, fully containerized — built for unstable internet, with English, Marathi, and Hindi support.

17 / AGRICULTURE / SUPPLY CHAIN TRACEABILITY

# QR & Blockchain Traceability Platform

Two-tier QR traceability down to the individual banana, with an optional blockchain seal that lets a buyer verify a shipment without taking the exporter’s word for it.

Client: **Banana export operation, Western India**

Agriculture | Supply Chain | Blockchain | QR Traceability | Polygon

PIPELINE — VERTICAL FLOW (BLOCKCHAIN-SEALED PATH; NO-BLOCKCHAIN MODE SKIPS 02-03)

- IN **Shipment Assigned** *every QR serial assigned before harvest*
- 01 **Confirmed at Load** *linked to shipment only once loaded*
- 02 **Merkle Tree Build** *every serial + truck record hashed*
- 03 **On-Chain Seal** *only the root hash written, one tx per truck*
- 04 **On-Scan Resolve** *pulls the full farm-to-store history*
- 05 **Merkle Proof Check** *recomputed and checked vs. on-chain root*
- OUT **Verified / Flagged** *any mismatch flagged as tampered*

## PROBLEM

A buyer receiving a shipment had no way to verify where it came from, or whether the harvest record behind it was genuine — only the exporter’s word for it. Making that verifiable meant either trusting one company’s database completely, or paying a blockchain transaction fee per banana, which was never going to be economical at real volume.

## OUR SOLUTION

**A two-tier QR traceability system, with an optional blockchain seal that removes the need to trust the exporter at all.** Every banana and every box carries a unique QR code, assigned to a shipment before harvest and confirmed only when actually loaded. Scanning either QR pulls the full farm-to-store history. For buyers who need independent proof, one cryptographic fingerprint covering an entire truckload is sealed on a public blockchain — one transaction per truck, not per banana — so a tampered record gets flagged automatically without a recurring per-unit blockchain cost.

## TECHNICAL DETAILS

### Serialized Two-Tier QR

Every banana sticker and shipping box carries a manufacturer-preprinted, unique serial QR code, assigned to a shipment record in advance and confirmed only when actually loaded onto the truck.

### Merkle-Sealed Blockchain Verification

Every banana serial range, box serial, and truck record is hashed into a Merkle tree; only the root hash is written on-chain, one transaction per truck rather than per banana or box.

### On-Scan Verification

Scanning a QR resolves the serial to its box and truck and displays the full history. For sealed shipments, the app recomputes a Merkle proof and checks it against the on-chain root.

### Stack

Polygon for low-cost, EVM-compatible on-chain anchoring; PostgreSQL as the off-chain store, with the blockchain holding only the tamper-proof fingerprint, never the underlying data.

# Poultry ERP & WhatsApp Ordering Platform

An ERP, CRM, and WhatsApp ordering system that replaces three notebook checkpoints and a retail till with one live, auditable ledger.

Client: **Poultry processing & retail company**

Food Processing | Poultry | WhatsApp AI | Multilingual | Traceability

PIPELINE — TWO-COLUMN VERTICAL

- IN — **Vendor Intake** — *birds + live weight, mobile form*
- 01 — **Slaughter Log** — *carcass weight + weight grade*
- 02 — **Yield Audit** — *carcass % vs. reference table by grade*
- 03 — **Cutting & Cold Store** — *SKU weight, order or cold store*
- 04 — **WhatsApp Order Match** — *B2B LLM-parsed / D2C menu flow*
- 05 — **Fulfillment Routing** — *nearest outlet, resolved from location*
- 06 — **Retail Sale** — *logged at till into central ledger*
- OUT — **Central Ledger** — *batch-traceable, dashboards, Excel export*

## PROBLEM

Three separate paper notebooks tracked intake, slaughter, and cutting, with a retail till logging sales on top of that. Orders arriving over WhatsApp — from contract clients and walk-in customers alike — were free text with no structure, no yield accountability, and no way to trace a finished order back to the batch it came from.

## OUR SOLUTION

**An ERP, CRM, and WhatsApp ordering system covering intake through order fulfillment as one system.** B2B orders are parsed by an LLM directly from nightly WhatsApp messages, matched to a registered contract by phone number, and run on a credit account; D2C orders go through a menu-driven WhatsApp flow that only finalizes once a Razorpay payment webhook confirms it. Every batch is yield-audited automatically, every order stays traceable back to its vendor batch, and incoming orders are routed to whichever outlet is physically closest to the customer.

## TECHNICAL DETAILS

### Multi-Channel WhatsApp Ordering

B2B messages are matched to a registered contract by phone and parsed by an LLM from code-mixed Hindi/Bengali/English text; a low-confidence parse gets a clarifying question instead of a guess. D2C orders finalize only once a Razorpay webhook confirms payment.

### Yield Auditing & Full Traceability

Every batch produces a carcass and cut yield %, checked against an expected-yield reference table by weight grade. The deviation threshold is set centrally by the dev team, never adjustable by owner or staff.

### Location-Aware Fulfillment

A customer’s location is resolved from a WhatsApp pin, a pasted Maps link, or a geocoded address, and the system proposes the nearest outlet, with staff able to override.

### Multi-Outlet & Role-Based Access

Outlets keep their own inventory, till, and vendor relationships rolling into one ledger. Four roles — Owner, Accountant, Facility, Retail — are enforced across every screen.

### Owner Dashboard

Nine live pages give a single real-time view across the business, with every dataset exportable to Excel.

# Resident Experience & Operations Platform

A resident experience that lives entirely in WhatsApp, feeding a single operations system built for the staff running the business.

Client: **Co-living operator, 60+ properties, India**

PropTech | Co-Living | WhatsApp AI | Role-Based Ops | Fintech

PIPELINE — VERTICAL FLOW, THE RESIDENT'S OWN JOURNEY

- IN **WhatsApp Entry** *room discovery via RAG chatbot, Hindi/Hinglish*
- 01 **ID Verification** *uploaded, grace period before move-in*
- 02 **Bed Assignment** *automatic, staff can override*
- 03 **Rent Payment** *UPI/Razorpay, confirmed by webhook only*
- 04 **Essentials Ordering** *photo catalogue, tracked warehouse handoff*
- 05 **Job Applications** *employer postings, consent-gated sharing*
- 06 **Issue Reporting** *structured, routed to the right role*
- OUT **Ops Dashboard** *9 role-scoped views, real-time sync*

## PROBLEM

More than 60 properties were coordinated through WhatsApp threads and Google Sheets, with no connection between resident-facing activity and how staff tracked it — a model already breaking down at the current count.

## OUR SOLUTION

**A resident experience that lives entirely in WhatsApp, feeding one operations system for the staff running the business.** A resident discovers a room, verifies ID, pays rent, orders essentials, applies to jobs, and raises issues without leaving a WhatsApp thread — updating the same system nine staff roles see in real time.

## MODULES

- **Foundation & Setup** — room/bed model, roles, policy content
- **Onboarding** — WhatsApp discovery, RAG FAQ, auto bed assignment
- **Payments** — UPI/Razorpay, reminders, auto-deduction
- **Essentials** — photo catalogue, NL search, fixed pickup
- **Work** — WhatsApp postings, consent-gated sharing
- **Support** — structured reporting, WiFi help, behaviour log

## TECHNICAL DETAILS

### Stack

Python/FastAPI + PostgreSQL, Redis + Celery, Docker; Node.js/TypeScript React; WhatsApp Business API; Ola Maps for room discovery.

### WhatsApp-Native Resident Experience

Every interaction runs inside one WhatsApp thread, Hindi or Hinglish, with a RAG chatbot handling FAQs. Bed assignment is automatic, and a resident can move in with ID still pending, given a grace period.

### Role-Scoped Internal Operations

Nine roles see only what's relevant to their scope — a property manager sees their property, Operations/Finance the whole portfolio. Live queues cover bookings, KYC, feedback, move-out.

### Payments & Traceability

A payment is confirmed only once a processor webhook verifies it, never a client-side message. Every handoff requires the receiving party to confirm quantity and condition.

### Privacy Built In

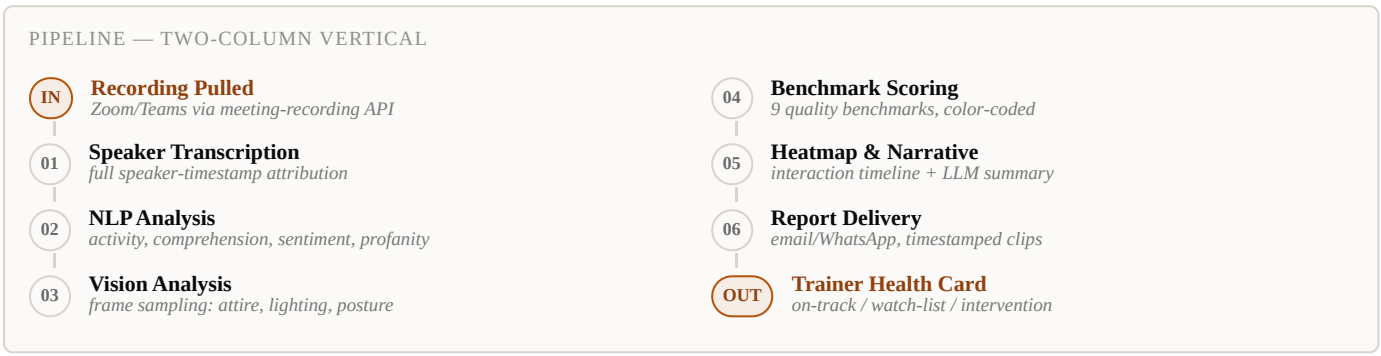
Consent is recorded separately per purpose. Documents are encrypted; an internal-only behaviour record is viewable only by the two most senior roles.

# AI Program Management Platform

A platform that automatically scores every live training session and manages instructor availability before a cancellation becomes a crisis.

Client: **Live-training delivery company**

EdTech | Live Training | Video Intelligence | WhatsApp AI | Program Management




## PROBLEM

Program managers reviewed every trainer's 3–5 hour live sessions manually to catch quality problems — disengagement, poor interaction, late starts — that often went unnoticed until real damage was done. A senior PM was spending roughly 72 hours a month just watching sessions, and last-minute instructor cancellations were often flagged too late to arrange a replacement.

## OUR SOLUTION

**A platform that automatically scores every live training session and manages instructor availability before a cancellation becomes a crisis.** Every session's audio, video, and chat is evaluated against nine quality benchmarks and rolled into a weekly per-trainer report the PM can review in minutes instead of hours. In parallel, a WhatsApp bot checks in with every instructor days ahead of their session, and the moment someone goes unavailable, it automatically pulls eligible replacements and hands the PM a ranked, one-tap confirmation list.

## TECHNICAL DETAILS

### Multi-Modal Session Scoring

Audio/video pulled via a meeting-recording API, transcribed with speaker-timestamp attribution, run through NLP for activity/sentiment/profanity, alongside a vision pipeline sampling frames for attire, lighting, posture, and camera-on status.

### Weekly Reporting

A conversation-heatmap renderer visualizes trainer-student interaction, and an LLM generates a plain-English narrative highlighting each trainer's strongest moment and biggest area to improve, with timestamped clips for any flagged benchmark.

### WhatsApp Availability Management

Automated check-ins three, two, and one day before a session; silence is treated as unavailability. A cancellation triggers an automatic query for eligible replacements and a ranked confirmation list back to the PM.

### Interface & Integrations

Google login, automatic roster/session sync to Google Sheets, color-coded trainer health cards with one-click drill-down, fully containerized deployment.

# CAD-to-Render Visualization Platform

A pipeline that turns a CAD floor plan directly into a full set of photorealistic marketing renders.

Client: **Mid-sized interior design firm**

Interior Design | CAD Automation | Generative AI | Architectural Visualization


## PROBLEM

Handed a raw CAD floor plan, an end client sees lines, labels, and geometry — not a home, and not something they can picture themselves living in. The old fix was a 3D animation or walkthrough built by hand for every plan, and even after that effort, it still looked stiff and artificial, never like a real photograph. And that manual process had to start over from scratch for every new layout and every revision.

## OUR SOLUTION

**A way to turn a CAD floor plan straight into photorealistic images that actually look like real photos of a finished space.** Instead of hand-building a 3D model and animating a walkthrough that still ends up looking artificial, the floor plan is read automatically and turned into a full set of finished renders — a hero shot, an overhead view, a full-house cutaway, a close-up of the best room, and an evening shot — each one convincing enough to use in real marketing. And because it’s automated, the team can now easily try different angles, lighting, shading, and furniture styles for the same plan, instead of starting the manual 3D process over from scratch every time.

## TECHNICAL DETAILS

### CAD Ingestion & Semantic Summarization

A DWG floor plan is converted to DXF and parsed into structured entity data, then compressed by a dedicated summarizer from a raw dump that can run past 100,000 characters down to a 2-4k character summary an LLM can actually reason about.

### Topology-Preserving Geometry Description

An LLM turns that summary into an 8-12 sentence geometry description under an explicit instruction to preserve the plan’s actual topology rather than let the model take creative liberties with room placement.

### Five Engineered Rendering Prompts Per Plan

The same pass produces five fully-specified photorealistic prompts covering a hero interior perspective, a top-down plan view, an isometric cutaway, a room-specific close-up, and a twilight shot — each with materials, lighting, and framing specified.

### Automated Rendering with Resumable Batches

An image-generation model renders each prompt automatically, with retry and exponential backoff on failure and a skip-if-already-generated check, so a batch can be queued and left to process without babysitting.

22 / HOSPITALITY / BRAND & WEB DESIGN

# Boutique Island Hostel Website

A photography-led website built entirely around the hostel’s own personality, not a generic travel-booking template.

Client: **Boutique hostel, Atlantic islands**

Hospitality | Travel | Web Design | Brand Identity | Photography-Led


## PROBLEM

A hostel this specific — sunsets, surf, a tiny village on a volcanic island — needed a website that felt like the place itself, not a generic travel-booking template. Every stock hostel site looks the same: a carousel, a booking button, and none of the actual character that makes someone want to go.

## OUR SOLUTION

**A photography-led website built entirely around the hostel’s own personality — the pink, the beach-town warmth, the specific island it sits on.** A full-bleed hero photo and a hand-drawn wordmark open the site before a single word of copy is read. Every space the hostel actually offers gets its own card with real photography instead of a stock icon. A loose, sun-bleached palette built around the brand’s signature pink, and a casual, first-person voice throughout make the site read like the people who actually run the place wrote it, not a template.

## DESIGN

### Photography-First Layout

Every section is built around real photos of the property — a full-bleed hero, a masonry-style gallery, and a card-based “Explore the Place” layout — so browsing the site feels like walking through the hostel.

### Brand-Led Color & Type

The hostel’s signature pink anchors the palette, paired with a soft yellow used only for calls to action, against clean white space and a rounded, friendly sans-serif.

### Voice Over Polish

Copy throughout — crew bios, FAQ answers, section headings — reads like the actual people running the hostel wrote it: casual, funny, specific to island life.

### Built to Browse, Not Just Book

Booking, WhatsApp, and Instagram are all one tap away, but the site’s real estate is spent on the things that actually sell a stay.