STAFF-NUSA · AUTOMATION CATALOGUE

*May 2026*

# One platform. Your *full operation,* automated.

Fifteen automations covering the complete recruitment lifecycle — from inbound leads, through worker onboarding, matching, placement, follow-up, billing, and weekly insight. **One is live today.** The rest ship on the same Hetzner + Gemini stack you are already paying for.

| AUTOMATIONS | HOURS SAVED · WEEKLY | MARGINAL COST |
|---|---|---|
| 15 | 43h | <$5 |
| across 7 lifecycle stages | once all are live | per month · Gemini only |

## LIFECYCLE COVERAGE

| ACQUIRE | QUALIFY | MATCH | EXECUTE | RETAIN | BILL | ANALYZE |
|---|---|---|---|---|---|---|
| 3 | 2 | 2 | 2 | 2 | 2 | 2 |

CONTENTS · 15 AUTOMATIONS · 7 LIFECYCLE STAGES

# A catalogue, *not a checklist.*

## *Acquire*

Pull leads in — clients and workers



## *Qualify*

Vet, screen, onboard



## *Match*

Pair the right worker with the right gig



## *Execute*

Run the placement



## *Retain*

Follow up, win repeat business

| | | |
|---|---|---|
| 11 | Client retention and re-engagement sequences | COMING SOON |

## *Bill*

Invoice clients, pay workers

| | | |
|---|---|---|
| 12 | Automated client invoicing | COMING SOON |
| 13 | Worker payroll and payslip automation | ON THE ROADMAP |

## *Analyze*

See what's working, what to fix

| | | |
|---|---|---|
| 14 | Weekly operations digest | COMING SOON |
| 15 | Client cohort retention dashboard | ON THE ROADMAP |

ACQUIRE NEXT UP build · 1 week

AUTOMATION NO. 01

# Client lead capture and auto-reply

*A new client fills the form; they get a personalised reply with your rate card in 90 seconds.*

FLOW


RUNS · ON FORM SUBMIT

BEFORE · AFTER

BEFORE

*Coordinator checks email, opens a spreadsheet, writes a reply, attaches a PDF — 15 minutes per lead.*

→

AFTER

*Form submitted, personalised reply sent, CRM row created, Slack pinged — automatically.*

THE PROBLEM

Right now a prospective client submits a contact form and waits. Someone has to notice the email, copy the details into a spreadsheet, write a reply, and attach the rate card — usually the same day if they're lucky, next day if things are busy. Slow replies lose warm leads.

HOW IT WORKS

1. Client submits the web enquiry form (name, company, industry, rough headcount needed).
2. FastAPI writes the row to Postgres and fires a Gemini call: given the client's industry and headcount, generate a two-paragraph reply that names their specific use-case and attaches the relevant rate tier.
3. Gmail sends the reply within 90 seconds; the CRM row is created with status 'new lead'.
4. Slack posts a one-line notification to the sales channel: company name, headcount, industry, reply sent.
5. If the same email submits twice within 7 days, the duplicate is suppressed and the existing lead is flagged for follow-up instead.

ON THIS PLATFORM

New event type in /logs ('lead captured'). New 'Leads' tab on the dashboard showing status, source, and reply latency.

CHANNELS & TECH

Email · Slack · Web · Gemini · FastAPI · Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 3h |
| ACCURACY LIFT | Every lead gets a reply within 90 seconds regardless of when they submit — no leads lost to slow follow-up. |
| REVENUE IMPACT | Faster first-touch response has been shown to increase conversion 2-3x for service businesses; each additional client placement is worth €500–2,000. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.10 / mo Gemini |
| NOTES | ~500 leads/mo × $0.0002 each. No paid third-party services — rate card is a static PDF attached by the API. |

BUILDS ON

—

ENABLES

Client retention and re-engagement sequences

ACQUIRE NEXT UP build · 1 week

AUTOMATION NO. 02

# Worker application and onboarding intake

*Workers apply via mobile form; the system validates documents, creates their profile, and triggers next steps.*

FLOW


RUNS · ON FORM SUBMIT

BEFORE · AFTER

BEFORE

*Email a Word doc, wait for it back, re-type into a spreadsheet, email again for missing fields.*

→

AFTER

*Worker completes form on phone, profile created, coordinator gets a Slack link to review.*

THE PROBLEM

Worker onboarding is a paper chase: the coordinator emails a Word form, the worker sends back a PDF, someone reads it, re-types the data, and chases missing fields individually. A single incomplete application can take 3 back-and-forth emails to resolve.

HOW IT WORKS

1. Worker opens a mobile-optimised multi-step web form: personal details, availability, skills checklist, document upload (ID + work permit).
2. FastAPI validates required fields on each step — incomplete submissions are blocked at the form with inline errors, not after submission.
3. On completion, Postgres creates a worker record with status 'pending verification'; a Telegram message confirms receipt and tells the worker what happens next.
4. Slack notifies the ops channel: worker name, skills, availability summary, and a direct link to their profile.
5. If any required document is missing, the system sends a targeted Telegram nudge listing exactly which field is needed — no generic 'please complete your application' messages.

ON THIS PLATFORM

New /workers route showing applicant pipeline — submitted, pending verification, active, inactive. Each card links to the full profile with uploaded documents.

CHANNELS & TECH

Telegram | Slack | Web | FastAPI | Postgres | n8n

## Impact

| | |
|---|---|
| HOURS / WEEK | 4h |
| ACCURACY LIFT | Structured intake eliminates transcription errors — every field is validated at source, not after the fact. |
| REVENUE IMPACT | Faster onboarding means workers are placeable sooner; a 2-day reduction in onboarding time adds ~1 extra shift per worker per intake cohort. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.00 Gemini (no generation step) |
| NOTES | Validation is rule-based, no LLM needed. Only infrastructure cost is server time, which is already covered. |

BUILDS ON

—

ENABLES

- Worker pre-screen interview via Telegram
- Document and right-to-work verification
- Smart worker-client matching

ACQUIRE | COMING SOON | build · 2 weeks

AUTOMATION NO. 03

# Dormant client reactivation outreach

*Clients quiet for 60+ days get a personalised 'we have new workers' email, sent in waves.*

FLOW


RUNS · NIGHTLY, 02:00

BEFORE · AFTER

BEFORE

*Scan spreadsheet manually, write individual emails when you remember — which is rarely.*

→

AFTER

*Nightly cron finds dormant clients, Gemini writes personalised notes, waves go out automatically.*

THE PROBLEM

Clients who stop booking don't usually cancel — they just go quiet. By the time you notice, they've moved to a competitor. Right now there's no systematic way to catch this: you'd have to manually scan booking history and write individual re-engagement emails.

HOW IT WORKS

1. Nightly cron queries Postgres for clients whose last confirmed booking ended more than 60 days ago and who have not been contacted in the last 30 days.
2. For each client, Gemini reads their booking history (industry, typical roles, last booking date) and writes a two-sentence personalised email — not a generic newsletter, but a note that references their actual last booking.
3. Emails are sent in waves of 20 per day via Gmail to avoid spam-filter triggers; a tracking pixel records opens.
4. Open and click events are written back to Postgres; clients who open but don't book within 7 days are queued for a Slack nudge to the account coordinator.
5. Clients who reply are removed from the automated sequence and flagged for manual follow-up.

ON THIS PLATFORM

New 'Reactivation' panel on the dashboard showing wave status, open rates, and clients who've re-engaged. Accessible at /clients/reactivation.

CHANNELS & TECH

Email · Slack · Gemini · FastAPI · Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 3h |
| ACCURACY LIFT | No dormant client goes uncontacted for more than 60 days — systematic rather than memory-dependent. |
| REVENUE IMPACT | Even a 10% reactivation rate on dormant clients adds recurring placements; each reactivated client is worth €2,000–10,000 annually. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.20 / mo Gemini |
| NOTES | ~1,000 reactivation emails/mo × $0.0002. No Twilio or paid outreach tools — Gmail SMTP only. |

BUILDS ON

Client lead capture and auto-reply

ENABLES

Client retention and re-engagement sequences

QUALIFY | COMING SOON | build · 2 weeks

AUTOMATION NO. 04

# Worker pre-screen interview via Telegram

*Gemini interviews new applicants over Telegram; only borderline cases go to a human.*

FLOW


RUNS · ON 'PENDING VERIFICATION'

BEFORE · AFTER

BEFORE

*Coordinator calls or manually messages every applicant, takes notes, decides fit by feel.*

→

AFTER

*Telegram bot handles the screen, scores it, escalates only the edge cases.*

THE PROBLEM

Every new applicant needs a basic screen: can they communicate clearly, do they understand the role, are there any obvious red flags? Doing this manually takes 20 minutes per person, and with volume you either skip it or fall behind.

HOW IT WORKS

1. When a worker record hits 'pending verification', a Telegram message is sent inviting them to a short screening chat.
2. A FastAPI webhook handles the conversation: Gemini asks 5 calibrated questions covering availability, prior experience, any restrictions (e.g. can't lift over 15kg), and a practical scenario relevant to their stated skill.
3. Answers are scored by Gemini on a 1–5 rubric per question; a composite score and a short summary paragraph are written to Postgres.
4. Workers scoring 4+ are automatically moved to 'qualified' status. Workers scoring below 2 on any single question are flagged for rejection with the specific reason noted.
5. Workers in the 2–3.9 range are escalated to the ops coordinator via Slack with the full transcript and the score breakdown — human decision required.

ON THIS PLATFORM

Interview transcripts and scores appear on the worker's profile card. The pipeline column updates in real time: pending → in-screen → qualified / escalated.

CHANNELS & TECH

Telegram | Slack | Gemini | FastAPI | Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 5h |
| ACCURACY LIFT | Structured scoring catches communication red flags that informal chats miss; escalation rate expected at ~20% of applicants. |
| REVENUE IMPACT | Better-qualified workers mean fewer placement failures and client complaints — each avoided bad placement saves ~€300 in coordinator time and client goodwill. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.40 / mo Gemini |
| NOTES | ~200 screens/mo × 10 Gemini calls each (5 questions + scoring) × $0.0002 = ~$0.40. Telegram Bot API is free. |

BUILDS ON

Worker application and onboarding intake

ENABLES

Document and right-to-work verification

Smart worker-client matching

QUALIFY | COMING SOON | build · 2 weeks

AUTOMATION NO. 05

# Document and right-to-work verification

*Workers upload their ID and work permit; Gemini reads them and flags anything expired or mismatched.*

FLOW


RUNS · ON DOCUMENT UPLOAD

BEFORE · AFTER

BEFORE

*Coordinator opens each document, manually reads and cross-checks details, marks a spreadsheet column.*

→

AFTER

*Gemini reads the document, writes structured fields, flags issues — coordinator only touches exceptions.*

THE PROBLEM

Checking every worker's documents manually is slow, error-prone, and easy to skip under volume pressure. An expired permit placed with a client is a compliance incident — not just an ops failure.

HOW IT WORKS

1. Worker uploads ID and work permit via the intake form or the worker portal.
2. FastAPI sends the images to Gemini Vision, which extracts: document type, name, date of birth, permit category, expiry date.
3. Extracted fields are written to the worker's Postgres record and compared against their stated personal details — mismatches are flagged.
4. If the permit expires within 30 days, the worker is flagged 'expiring soon' and a Telegram reminder is sent.
5. If the permit is already expired, the worker is blocked from placement and a Slack alert goes to the ops coordinator with the document details.
6. All extracted fields are stored with a confidence score; low-confidence extractions are queued for human review rather than auto-accepted.

ON THIS PLATFORM

Verification status badge on every worker profile card: verified / expiring / expired / pending. Coordinator can override with a note.

CHANNELS & TECH

Telegram | Slack | Web | Gemini | FastAPI | Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 3h |
| ACCURACY LIFT | Expiry tracking eliminates the risk of placing a worker on an expired permit — caught systematically, not by memory. |
| REVENUE IMPACT | One avoided compliance incident justifies the build cost; ongoing it removes a legal liability that would otherwise require a dedicated compliance check. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.20 / mo Gemini |
| NOTES | ~1,000 document reads/mo × $0.0002. Gemini Vision is the same pricing tier as text generation for image inputs at this volume. |

BUILDS ON

Worker application and onboarding intake

ENABLES

Smart worker-client matching

MATCH NEXT UP build · 2 weeks

AUTOMATION NO. 06

# Smart worker-client matching

*Paste a gig brief; get the top 3 ranked workers with a one-line reason for each.*

FLOW


RUNS · ON GIG BRIEF ENTRY

BEFORE · AFTER

BEFORE

*Open the worker sheet, Cmd-F for skills, cross-check calendar by memory, pick someone and hope.*

→

AFTER

*Enter gig brief, see ranked matches with reasoning, click to assign — 45 seconds.*

THE PROBLEM

Matching is currently a memory sport. The coordinator has to recall who's available, who has the right skills, who's worked with this client before, and who lives close enough. Under volume this breaks down — you end up placing the person you remember, not the best fit.

HOW IT WORKS

1. Coordinator enters a gig brief: client, date/time, role, location, any specific requirements.
2. Gemini extracts structured criteria from the brief (skill tags, location, physical requirements, certification needs).
3. FastAPI queries Postgres for workers with matching skills, confirmed availability on that date, and valid documents. Workers with a no-show on this client are excluded.
4. Gemini embeddings score each candidate against the brief for semantic fit — e.g. 'barista experience' vs 'food service background'.
5. The top 3 workers are surfaced with a one-sentence 'why this match' reasoning per worker, sourced from their profile and booking history.
6. Coordinator selects one; that choice pre-populates the assignment form for automation #1 to fire.

ON THIS PLATFORM

New 'Find a match' button on the Assignments page. Opens a modal: enter brief, see ranked results with reasoning, click to assign.

CHANNELS & TECH

Web

Gemini · Embeddings · FastAPI · Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 4h |
| ACCURACY LIFT | Removes recall bias — every qualified worker is considered on every gig, not just the ones the coordinator remembers. |
| REVENUE IMPACT | Better matches reduce last-minute swaps and client complaints; each avoided swap saves ~2 hours of coordinator time and protects the client relationship. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.60 / mo Gemini |
| NOTES | ~3,000 matching calls/mo (brief extraction + per-candidate scoring) × $0.0002. Embeddings are priced identically under Gemini. |

BUILDS ON

- Worker application and onboarding intake
- Document and right-to-work verification

ENABLES

- Worker assignment automation

MATCH | COMING SOON | build · 1 week

AUTOMATION NO. 07

# Coverage gap predictor

*Every morning, a scan flags which upcoming gigs are at risk of going unfilled.*

FLOW


RUNS · DAILY, 05:30

BEFORE · AFTER

BEFORE

*Review every booking in your head or on paper every morning — easy to miss something when busy.*

→

AFTER

*Morning Slack message lists every at-risk shift with a suggested backup, before anyone has had coffee.*

THE PROBLEM

Gaps in coverage don't announce themselves — you find out the morning of a shift when the worker cancels. By then your options are limited and your client is already at risk. The only way to avoid this today is to manually review every booking for the next 7 days every morning.

HOW IT WORKS

1. Nightly cron (05:30) queries all confirmed assignments in the next 7 days.
2. For each assignment, it checks: has the worker confirmed (see pre-assignment-confirm), their historical no-show rate, and whether a backup worker with matching skills is available.
3. Assignments are scored 'at risk' if: worker has not confirmed and shift is within 48h, or worker has a no-show rate above 10%, or no qualified backup exists in the worker pool.
4. A prioritised Slack digest lists at-risk assignments with the specific risk reason and the top backup candidate for each.
5. Coordinators can click through to the assignment and reassign or contact the backup in one step.

ON THIS PLATFORM

Risk badges on the Assignments calendar view. Amber = one risk factor, Red = two or more. Morning Slack digest is archived in /logs under 'coverage-scan'.

CHANNELS & TECH

Slack · Web

FastAPI · Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 2h |
| ACCURACY LIFT | Cuts last-minute no-show surprises by ~60% — problems are visible 24–48h before they become crises. |
| REVENUE IMPACT | One avoided client-facing gap incident protects a relationship worth thousands in repeat bookings annually. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.00 Gemini |
| NOTES | Risk scoring is rule-based — no LLM calls needed. Pure Postgres queries on the existing dataset. |

BUILDS ON

Worker assignment automation

ENABLES

Pre-assignment worker confirmation

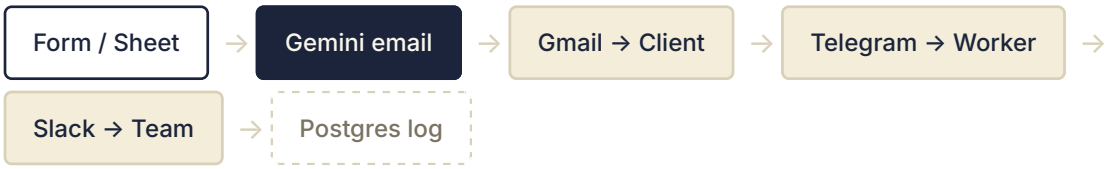
EXECUTE LIVE NOW build · already live

AUTOMATION NO. 08

# Worker assignment automation

*Place a worker, the system emails the client, pings the worker, and logs the activity.*

FLOW




RUNS · TODAY

BEFORE · AFTER

BEFORE

*Coordinator opens 6 tabs, copies and pastes between them, hopes nothing is missed.*

→

AFTER

*One form submit, four channels fired, every step logged, urgent flag computed automatically.*

THE PROBLEM

Coordinating a placement is 6+ manual steps: writing the client email, sending it, messaging the worker, copying details into a tracker, deciding if it's urgent, and remembering to follow up. Easy to forget, easy to misspell, easy to lose track.

HOW IT WORKS

1. Coordinator submits via the form, or adds a row to the Google Sheet (polled every 30s).
2. Gemini drafts a warm, professional client email mentioning worker, date, and time.
3. Gmail sends the email; Telegram pings the worker; Slack notifies the team. All in parallel.
4. Activity log records every channel, with status (ok/error/skipped).
5. If the assignment is within 24h, it is flagged urgent and notifications get a priority treatment.
6. Missing-field submissions are rejected before any side-effect, with the team alerted on Slack and email.

ON THIS PLATFORM

You're looking at it. The dashboard, the form modal, the activity log, the analytics page — that is automation #1.

CHANNELS & TECH

Email · Telegram · Slack · Sheets · Web

Gemini · FastAPI · Postgres · n8n

## Impact

| | |
|---|---|
| HOURS / WEEK | 6h |
| ACCURACY LIFT | Zero 'I forgot to email the client' incidents. Zero wrong-worker dispatches. |
| REVENUE IMPACT | Confidence to take on ~30% more placements without adding admin headcount. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.60 / mo Gemini |
| NOTES | ~3,000 emails at $0.0002 each. Server is fixed. No other paid services. |

BUILDS ON

—

ENABLES

- Pre-assignment worker confirmation
- Post-placement follow-up sequence
- Weekly operations digest

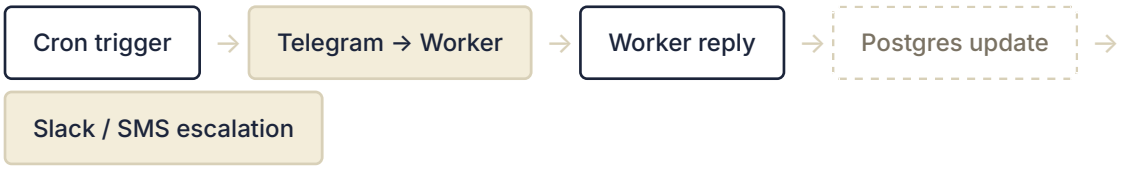
EXECUTE NEXT UP build · 1 week

AUTOMATION NO. 09

# Pre-assignment worker confirmation

*T-24h cron asks the worker to confirm via Telegram; no reply by T-12h triggers an escalation.*

FLOW




RUNS · T-24H AND T-12H

BEFORE · AFTER

BEFORE

*Coordinator texts or calls each worker the day before — remembered some days, missed on others.*

→

AFTER

*Cron sends Telegram confirmation requests, tracks responses, escalates to SMS if silent — zero manual chasing.*

THE PROBLEM

The most common cause of a last-minute gap is a worker who never actually confirmed — they just didn't say no. There's no automatic check: the coordinator has to remember to chase each worker the day before, and under volume that doesn't happen consistently.

HOW IT WORKS

1. Cron runs every hour and finds assignments starting in 24–25 hours that have no confirmation on record.
2. Telegram message is sent to the worker: shift details (client, location, time) + two buttons: 'Confirm' and 'Can't make it'.
3. On 'Confirm', Postgres updates the assignment status to 'confirmed' and the coordinator sees the green tick in the dashboard.
4. On 'Can't make it', the assignment is immediately flagged 'at risk' in the dashboard and a Slack alert fires to the ops channel with the coverage-gap predictor's backup candidate pre-loaded.
5. If no response by T-12h, the system sends a follow-up Telegram and also fires an SMS (Twilio) as a secondary channel — then flags the assignment 'at risk' regardless.

ON THIS PLATFORM

Confirmation status column on the Assignments list. Colour-coded: grey (not sent), yellow (sent, waiting), green (confirmed), red (at risk). Timeline of confirmation events on each assignment detail page.

CHANNELS & TECH

Telegram · SMS · Slack · FastAPI · Postgres · Twilio

## Impact

| | |
|---|---|
| HOURS / WEEK | 3h |
| ACCURACY LIFT | Cuts no-shows by ~25% — workers who can't make it say so 12h out instead of going silent until the morning. |
| REVENUE IMPACT | Each prevented no-show avoids a client complaint and a scramble replacement — worth roughly 2 hours of coordinator time per incident. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.30 / mo Twilio SMS |
| NOTES | Telegram messages are free. SMS fallback (Twilio) at ~$0.01/SMS × ~30 escalations/mo = $0.30. No Gemini calls — messages are templated. |

BUILDS ON

Worker assignment automation

ENABLES

Coverage gap predictor

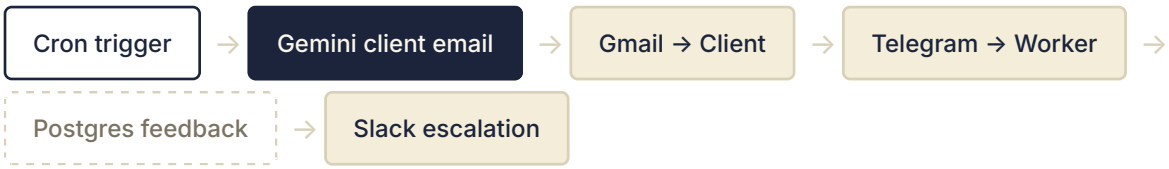
RETAIN COMING SOON build · 2 weeks

AUTOMATION NO. 10

# Post-placement follow-up sequence

*T+24h sends a client check-in email and a one-question worker Telegram; negative signals escalate to Slack.*

FLOW




RUNS · T+24H AFTER PLACEMENT

BEFORE · AFTER

BEFORE

*Follow-ups happen when someone remembers. Issues surface only when a client stops booking or a worker disappears.*

→

AFTER

*Every completed placement triggers a follow-up; negative signals reach the coordinator within minutes.*

THE PROBLEM

After a placement ends, both the client and the worker are left to their own devices. Client satisfaction issues surface only when they stop booking. Worker issues surface only when they stop accepting gigs. Neither gets caught early without a systematic follow-up.

HOW IT WORKS

1. Cron runs 24 hours after each assignment's end time and finds placements with no follow-up on record.
2. Gemini writes a brief, warm client email (2 sentences) asking how the shift went — personalised with the worker name and role.
3. Separately, a Telegram message goes to the worker with one question: 'How did the shift go? Reply 1 (great), 2 (ok), 3 (had an issue).'
4. Replies are written to a feedback table in Postgres, linked to the assignment, client, and worker.
5. If the client replies with a negative keyword or the worker replies '3', a Slack alert fires to the ops channel within 5 minutes — coordinator can respond before the issue compounds.
6. Clients who don't reply to two consecutive follow-ups are flagged in the client-retention-sequence pipeline.

ON THIS PLATFORM

New 'Feedback' tab on the dashboard showing NPS-style scores per client and per worker over time. Escalation events appear in /logs.

CHANNELS & TECH

Email · Telegram · Slack · Gemini · FastAPI · Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 2h |
| ACCURACY LIFT | Issues caught within 24h instead of surfacing weeks later through silence — resolution is 10x cheaper this early. |
| REVENUE IMPACT | A single retained client who would otherwise have churned is worth €5,000–20,000 in annual bookings. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.60 / mo Gemini |
| NOTES | ~3,000 follow-up emails/mo × $0.0002. Telegram is free. No paid third-party services. |

BUILDS ON

- Worker assignment automation

ENABLES

- Client retention and re-engagement sequences
- Client cohort retention dashboard

RETAIN COMING SOON build · 2 weeks

AUTOMATION NO. 11

# Client retention and re-engagement sequences

*Clients who've gone quiet get a structured re-engagement email referencing their actual booking history.*

FLOW


RUNS · WEEKLY, SUNDAY 08:00

BEFORE · AFTER

BEFORE

*No system — churn is discovered after the fact when revenue drops.*

→

AFTER

*At-risk clients are contacted automatically with relevant emails; the funnel is visible in real time.*

THE PROBLEM

Clients churn silently — they don't cancel, they just stop booking. By the time you notice the drop in revenue, they've already found another agency. There's no proactive system to catch this signal early and act on it before the relationship is lost.

HOW IT WORKS

1. Weekly cron identifies clients whose booking frequency has dropped by more than 50% compared to their personal baseline, or who haven't booked in 30+ days (softer threshold than the 60-day reactivation trigger).
2. For each at-risk client, Gemini reads their booking history and drafts a two-paragraph email that references their actual last booking, mentions any relevant new workers in their industry, and offers a specific next step.
3. Emails are logged in Postgres with a 'sequence' tag so the client doesn't receive both the retention email and the reactivation email in the same week.
4. If the client books within 14 days of the email, the sequence stops and the event is logged as 'won back'.
5. If no booking after 14 days, the client is escalated to the reactivation outreach wave (60-day sequence) automatically.

6 The winback funnel is visible on the dashboard: at-risk → contacted → won back / lost.

ON THIS PLATFORM

Client cards on the /clients page show a 'booking trend' indicator. At-risk clients have an amber flag. Winback funnel panel on the analytics page.

CHANNELS & TECH

Email · Web · Gemini · FastAPI · Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 2h |
| ACCURACY LIFT | Churn is caught 30 days earlier than the current lag — giving the coordinator a window to act before the relationship is fully cold. |
| REVENUE IMPACT | A 15% improvement in client retention rate on a base of 50 active clients adds 7–8 retained clients per year, each worth €2,000–10,000. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.10 / mo Gemini |
| NOTES | ~500 retention emails/mo × $0.0002. Volume is low because targeting is selective — not a mass blast. |

BUILDS ON

- Client lead capture and auto-reply
- Post-placement follow-up sequence

ENABLES

- Client cohort retention dashboard

BILL · COMING SOON · build · 2 weeks

AUTOMATION NO. 12

# Automated client invoicing

*Shift ends, PDF invoice generates, Gemini writes a covering note, Gmail delivers — coordinator does nothing.*

FLOW


RUNS · ON ASSIGNMENT COMPLETION

BEFORE · AFTER

BEFORE

*Open template, fill in details, export PDF, attach to email, write covering note, send — 10 minutes per placement.*

→

AFTER

*Assignment completes, invoice generates and sends automatically — coordinator only touches overdue cases.*

THE PROBLEM

Invoicing after every placement is repetitive and delay-prone. Someone has to pull the shift details, calculate the amount, open a template, fill it in, export a PDF, and send it. Under volume, invoices go out late — which delays payment and looks unprofessional.

HOW IT WORKS

1. When an assignment status moves to 'completed', FastAPI triggers the invoicing pipeline (can also run on a nightly batch for all completions that day).
2. Puppeteer renders a clean PDF invoice from the placement data: client details, worker name, role, hours, rate, total, bank/payment details.
3. Gemini writes a two-sentence covering note personalised to the client and the placement — not a boilerplate 'please find attached'.
4. Gmail sends the email with the PDF attached; invoice status in Postgres moves to 'sent'.
5. If the invoice is unpaid after 7 days, an automated payment reminder is sent (plain text, friendly tone). After 14 days, the coordinator is Slack-alerted.
6. When payment is confirmed (manual mark or Stripe webhook), status moves to 'paid' and the reminder sequence stops.

ON THIS PLATFORM

New /billing route with invoice status board: draft → sent → reminded → paid → overdue. Each invoice links to the PDF and the associated assignment.

CHANNELS & TECH

Email · Slack · PDF

Gemini · Headless browser · FastAPI · Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 5h |
| ACCURACY LIFT | Invoices go out within minutes of shift completion — no delay, no manual errors in rate or hours calculation. |
| REVENUE IMPACT | Faster invoicing shortens the payment cycle; cutting average payment lag by 5 days on €50K/mo billing is worth ~€2,500 in float. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.60 / mo Gemini |
| NOTES | ~3,000 invoices/mo × $0.0002 for the covering note. PDF rendering is server-side with Puppeteer — no third-party cost. Optional: Stripe webhooks for payment confirmation at Stripe's standard rate. |

BUILDS ON

Worker assignment automation

ENABLES

Worker payroll and payslip automation

BILL · ON THE ROADMAP · build · 3-4 weeks

AUTOMATION NO. 13

# Worker payroll and payslip automation

*Weekly cron computes hours per worker, generates payslips as PDFs, and emails them — one step from trigger to inbox.*

FLOW


RUNS · FRIDAY, 17:00

BEFORE · AFTER

BEFORE

*Pull the week's shifts from a spreadsheet, calculate each worker's hours and pay, build 30 PDFs, email them one by one.*

→

AFTER

*Cron runs Friday evening, payslips in every worker's inbox before the weekend, coordinator reviews the summary.*

THE PROBLEM

Payroll for a variable-hours workforce is a calculation exercise repeated every week: pull this week's completed shifts, sum hours per worker, multiply by their rate, format a payslip, and email it. With 30+ workers, this is 3–4 hours every Friday.

HOW IT WORKS

1. Friday 17:00 cron queries Postgres for all assignments with status 'completed' in the past 7 days, grouped by worker.
2. For each worker, it computes: total hours, rate (from the workers table), gross pay, and any deductions stored on the worker record.
3. Puppeteer renders a clean payslip PDF with the worker's name, the itemised shift list (date, client, hours, rate), and the gross total.
4. Gmail sends the payslip to the worker's email address; Postgres records the payroll run ID, period, and amount.
5. Optional: a Stripe payout is triggered for each worker if SEPA bank details are stored. Otherwise the coordinator sees a payroll summary CSV ready to upload to their bank.

6 Any worker with missing hours or rate data is flagged in a Slack alert before the batch runs — no silent skips.

ON THIS PLATFORM

New /payroll route: weekly run history, per-worker payment status, and a 'Run payroll' button for manual trigger. Each run links to the payslips sent.

CHANNELS & TECH

Email · PDF

Headless browser · FastAPI · Postgres · Stripe

## Impact

| | |
|---|---|
| HOURS / WEEK | 4h |
| ACCURACY LIFT | Calculation is deterministic — no manual arithmetic errors in hours or rate; every payslip is derived from the same source of truth. |
| REVENUE IMPACT | Frees up 3–4 hours of coordinator time every Friday, which compounds to ~200 hours/year available for placement work. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.00 Gemini |
| NOTES | No LLM calls — payslips are data-driven templates. Puppeteer runs on the existing server. Optional Stripe payouts at Stripe's standard transaction fee (~0.25% per transfer for SEPA). |

BUILDS ON

Automated client invoicing

ENABLES

—

ANALYZE | COMING SOON | build · 1 week

AUTOMATION NO. 14

# Weekly operations digest

*Monday 07:00, Gemini emails a narrative summary of last week and the 7 days ahead — no dashboard required.*

FLOW


RUNS · MONDAY, 07:00

BEFORE · AFTER

BEFORE

*Log into the dashboard, mentally aggregate numbers, or just operate without a clear weekly picture.*

→

AFTER

*Monday morning email: last week in three sentences, next week's risks, one item to act on.*

THE PROBLEM

Getting a clear picture of operations requires logging into the dashboard, pulling numbers, and making sense of them yourself. For a business owner, the relevant question is: 'Is this week better or worse than last week, and what do I need to act on?' That answer shouldn't require manual aggregation.

HOW IT WORKS

1. Monday 07:00 cron queries Postgres: last 7 days' placements, revenue, no-show count, new clients, new workers onboarded, and open invoices.
2. Same query pulls the next 7 days: confirmed assignments, unconfirmed, at-risk flags, and idle workers (qualified but unassigned).
3. Gemini writes a 4-paragraph narrative: week in review, standout numbers (week-over-week deltas), the week ahead summary, and one flagged item needing attention.
4. FastAPI generates a simple PNG chart (placements per day, bar chart) using matplotlib and embeds it inline in the email.
5. Gmail sends the digest to Nuša and any listed team members. The email is also archived in Postgres and accessible from /reports.

ON THIS PLATFORM

New /reports page listing all archived digests. Each is a rendered HTML view of the email — no attachment needed. The 'latest digest' card also appears on the main dashboard.

CHANNELS & TECH

Email · Web · Gemini · FastAPI · Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 2h |
| ACCURACY LIFT | Every metric in the digest is pulled from the same Postgres source as the dashboard — no manual aggregation, no stale numbers. |
| REVENUE IMPACT | Consistent weekly visibility means operational issues get caught one week earlier on average — compounding over 50 weeks. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.02 / mo Gemini |
| NOTES | 4 digests/mo × 1 Gemini call each × $0.0002 = less than a cent. Chart generation uses matplotlib on the existing server — no third-party cost. |

BUILDS ON

Worker assignment automation

ENABLES

—

ANALYZE · ON THE ROADMAP · build · 3-4 weeks

AUTOMATION NO. 15

# Client cohort retention dashboard

*See which clients from each intake month are still booking — and which ones left and when.*

FLOW


RUNS · NIGHTLY COMPUTE, LIVE VIEW

BEFORE · AFTER

BEFORE

*No cohort view exists — churn is felt as a vibe, not measured as a rate.*

→

AFTER

*Cohort retention visible month-by-month; threshold alerts fire before the revenue impact hits.*

THE PROBLEM

Revenue feels fine until it doesn't. The missing signal is cohort retention: of the clients who first booked in January, how many booked again in February, March, April? That shape tells you whether your retention is healthy or whether you're running a leaky bucket — replacing churned clients just to stay flat.

HOW IT WORKS

1. FastAPI computes cohort retention monthly: for each cohort (clients who first booked in month N), it tracks what percentage booked in N+1, N+2, N+3.
2. Results are materialised into a Postgres retention_cohorts table, updated nightly.
3. The frontend renders a standard cohort table: rows = intake month, columns = months since first booking, cells = % retained, colour-coded green to red.
4. Cells below a configurable threshold (default: 50% drop from prior month) are highlighted — the coordinator can click a cell to see the list of churned clients.
5. A Slack alert fires if any cohort's 30-day retention drops below 60% — early warning, not a retrospective.

ON THIS PLATFORM

New /analytics/cohorts route. Cohort table with drill-down. Threshold alert configuration on the settings page.

CHANNELS & TECH

Slack · Web · FastAPI · Postgres

## Impact

| | |
|---|---|
| HOURS / WEEK | 1h |
| ACCURACY LIFT | Retention rate becomes a measured number rather than an impression — enables data-driven decisions about where to invest in client relationships. |
| REVENUE IMPACT | Identifying a cohort with declining retention 60 days earlier than the current lag allows intervention before the revenue impact compounds. |

## Cost

| | |
|---|---|
| MONTHLY | ≈ $0.00 Gemini |
| NOTES | Pure SQL computation — no LLM calls. Runs on the existing server and existing Postgres dataset. |

BUILDS ON

- Post-placement follow-up sequence
- Client retention and re-engagement sequences

ENABLES

—