



Hiring that tells you what to do next.

A complete, AI-native recruiting desk — from the careers page a candidate applies on to the offer email you send. Résumés read themselves, candidates are scored against the real role, and every screen answers one question: **who is owed an action, and what is it?**

Résumé parsing • Gemini multimodal

Semantic candidate search

AI fit scoring with evidence

Interview scheduling + .ics

Talent bank, auto-filed

DPDP-aligned by design

01 THE PROBLEM

Every ATS stores applicants. Almost none of them tell you who to call.

A small team hiring for three roles doesn't drown in candidates — it drowns in *undecided* candidates. The cost isn't storage. It's the strong applicant who waited eleven days because nobody opened the PDF.

01

Résumés nobody has read

A folder of PDFs is not a pipeline. Someone still has to open each one, and that someone is always busy.

02

Screening by keyword

Filters reward whoever stuffed the most nouns into a skills list, and quietly bury the person who actually shipped the thing.

03

Status without instruction

"Reviewing · 14 candidates" is a number, not a next step. The dashboard describes the backlog instead of clearing it.

04

Silence as a default

Rejections go unsent because writing them is unpleasant. The candidate learns nothing; the brand pays for it.

02 HOW IT WORKS

One flow, from the apply button to the offer.

The candidate-facing careers page and the recruiter's cockpit are one system on one database. Nothing is exported, re-imported, or re-keyed between them.

CANDIDATE

Applies on your careers page

- Live roles publish straight from the admin — set a role to *Open* and it appears
- Résumé uploads to a **private** bucket, never a public URL
- Explicit consent + honeypot + per-IP burst and daily caps
- Submission is acknowledged instantly; AI work happens after

THE SYSTEM

Reads and understands it

- PDFs read natively — columns, tables and layout survive
- DOCX extracted, then structured
- Name, title, years, skills, education, summary pulled out
- A 768-dim embedding makes the pool **searchable by meaning**
- Parse failures are flagged and retryable, never silent

RECRUITER

Works a queue, not an inbox

- Score every unread résumé in one click
- Rank a role's candidates with the evidence attached
- AI-drafted invite, pass, offer or follow-up
- Interview booked, scored, decided
- Every résumé you've ever held, filed by role

The organising idea: a work queue, not a funnel.

Most hiring dashboards lead with stage counts. This one leads with “**who needs what, most urgent first.**” Each candidate resolves to exactly one recommended action — *Screen with AI, Invite to interview, Read and decide, Record the interview, Send a polite pass* — with a plain-English reason next to it.

The ordering reflects what actually costs the business: a strong candidate going cold outranks tidying up a weak one, and someone unread for a week is escalated above someone who applied this morning.

Seven surfaces. Each one closes a decision.

Everything below is built and running in production today.

/hr

Hiring cockpit

The page you open first. Action-first, never a wall of stage counts.

- **Work queue** — every candidate owed an action, ranked by urgency, with the AI's one-line "why them"
- **Interview schedule** — what your week looks like, lifted out of the email log automatically
- **Decision speed** — median days to answer a candidate, and the oldest still-open application
- **AI coverage** — how much of the pool has been read, score bands, parse failures
- **Inbox aging** — how long the oldest unreviewed application has been sitting
- **Talent read** — average experience and the most common skills in your pool
- **Per-role table** — applicants, fresh, strong matches and average score for every opening

/careers

Applicants list

The full pool, with server-side everything so it stays correct at any size.

- Filter by role, stage, AI verdict band, or free text on name and email
- Sort by newest, oldest, best score, worst score or name — real SQL ordering, correct across pages
- Every row carries its owed action, so the list is scannable as a to-do
- Exact server-side counts — the total always describes the rows you're paging through

/careers/[id]

Candidate workspace

One person, five tabs, nothing lost when you switch between them.

- **Score** — fit against a chosen role: 0–100, verdict, matched vs missing skills, and 2–4 concrete evidence points
- **Questions** — 6–8 screening questions written for *this* candidate and *this* role
- **Reply** — the AI email composer (see below)
- **Résumé** — the real file, rendered in-page from the private bucket
- **Profile** — parsed details, links, cover note, consent record
- **Repeat-applicant check** — every other application from the same email, so nobody is screened or rejected blind
- **Interview scorecard** + stage, rating and notes controls

/hr/shortlist

AI shortlist

"Who are the best five people for this role?" — answered with reasoning.

- Semantic recall over the whole pool first, then per-candidate LLM scoring
- Scope it to one role's applicants or search **every** résumé you hold
- Ranked with score, verdict, similarity and the evidence behind each placement
- Results are cached — landing on the page shows the last ranking instead of an empty box

/careers/jobs

Job openings

The roles that feed everything else, managed in one place.

- Full CRUD: title, department, location, type, level, description, responsibilities, requirements, skills, salary range, headcount
- Status drives the public site — **Open** publishes, **Draft** and **Closed** don't
- Saving pushes a revalidation to the site, so a new role is live in seconds rather than waiting out a cache window
- The built-in copilot can draft a complete opening and prefill the form

/careers/chat

Résumé chat

Ask the candidate pool a question in English.

- "Who has production RAG experience?" · "Compare the top three for the backend role" · "Anyone in Pune with 5+ years?"
- Grounded strictly in the résumés you hold — it cites candidates by name and refuses to invent them
- Context is refreshed every turn, so a deleted candidate stops being recommended immediately
- Scopeable to a single role

/hr/resume-bank

Résumé bank

The standing talent pool — referrals, events, LinkedIn finds, walk-ins.

- Upload a résumé and the AI **files it under the right open role by itself**, reading discipline and stack rather than seniority
- It extracts name, title, years, skills and a one-line summary while filing
- Bank uploads and careers applicants shown side by side, grouped under the same roles
- Re-file by hand any time; the AI's placement is marked as such until you override it
- **One-click export** to a ZIP — one folder per role plus a manifest CSV of everything the AI read
- Applicant résumés are read-only here, so the bank can never damage a live application

04 CANDIDATE COMMUNICATION

Drafted by AI. Sent by a human. Logged honestly.

Four kinds of reply — **interview invite**, **polite pass**, **offer**, **ask for details** — each written in your voice, grounded in that candidate's actual résumé.

What the drafts get right

- An invite names **one concrete thing** from the résumé that earned it — not a form letter
- A rejection is short and kind: no score, no list of weaknesses, no false promise to "keep you on file"
- An offer says yes warmly and defers terms to the formal document — it never invents salary or a joining date
- Dates and times are computed outside the model and handed to it as fixed text, because that is the one fact that must never be wrong
- You can steer any draft with a one-line note ("mention the RAG work", "offer Tuesday too")

Why sending stays manual

- The composer opens a **prefilled Gmail tab** from your own mailbox
- The reply lands where the thread already lives — your inbox, not a no-reply address
- Nothing is logged until you confirm you actually hit send, so the history records what a human witnessed rather than what software assumed
- Interview invites also generate a proper **.ics calendar file**, re-downloadable later
- Sending a mail advances the candidate's stage automatically — invite → interview, pass → rejected, offer → hired

05 UNDER THE HOOD

The AI is specific, cited, and cheap to run.

Six distinct model jobs, each with its own prompt, its own guardrails, and its own line on the cost dashboard.

JOB 01

Résumé understanding

PDFs go to a multimodal model whole, so multi-column layouts and tables survive intact. DOCX is extracted locally first. Output is structured JSON, not a text blob.

JOB 02

Semantic recall

pgvector over 768-dimension embeddings finds candidates by *meaning*. A search for "built retrieval systems" surfaces the person who never wrote the word "RAG".

JOB 03

Fit scoring

Scores a candidate against one role using only résumé evidence. Explicitly **penalises keyword-stuffing** — a skill listed with no supporting work does not earn credit.

JOB 04

Batch screening

Reads every unscored résumé in the pool sequentially, in durable batches, and persists both the score and the reasoning. A partial run is still useful and reports exactly what landed.

JOB 05

Interview questions

Verifies their strongest claims, probes gaps against the requirements, and adds behavioural fit — written for this pairing, not pulled from a bank.

JOB 06

Auto-filing

Reads a banked résumé against your open roles and files it by discipline and stack. Unreadable files park in the general pool for a human, never guessed at.

Reasoning is stored, not re-bought.

Every assessment is cached with its evidence, so opening a candidate shows why they scored 82 without paying to ask again. Every model call in the suite meters to a dedicated **HR** bucket, so the costing dashboard prices hiring as its own line rather than burying it in "other AI" — you can see what a hire actually cost you in inference.

06 PRIVACY & COMPLIANCE

Built for India's DPDP Act, not retrofitted to it.

Résumés are among the most sensitive documents a company holds. The architecture treats them that way — and the promises made to candidates are enforced in code, not in a policy page.

✓ Private storage, always

The résumé bucket has no public policy. Files are reachable only through a session-gated proxy or a short-lived signed URL. There is no shareable link to leak.

✓ Demonstrable consent

The form requires explicit consent, and the consent plus its timestamp is recorded on the candidate's record and shown next to their résumé — so it can be proven, not just claimed.

✓ Candidate data can't leak into your public chatbot

Résumé embeddings live in the applications table only and are queried only by the admin. They are never ingested into the public knowledge store, so a website visitor can't surface applicant data.

✓ Deletion is real deletion

Removing an applicant also removes their résumé file. No orphaned PII left behind in storage.

✓ No third-party document viewers

A .docx that a browser can't render shows the text already extracted, rather than shipping a candidate's résumé to an external preview service nobody told them about.

✓ No fully automated decisions

The AI scores, ranks and drafts. A human sets the stage, sends the mail and makes the call. The published recruitment-privacy notice states this to candidates.

✓ Abuse controls on intake

Honeypot field, server-side validation, file type and size limits, plus per-IP burst and daily caps on the public apply endpoint.

✓ Degrades honestly

When a data source is down, the affected panel says so rather than rendering a confident zero. Aggregates report how many records they actually scanned.

Modern, boring, and cheap to operate.

LAYER	WHAT IT IS
Frontend	Next.js App Router, server components, server actions. Deployed on Vercel.
Database	Postgres (Supabase) with pgvector . Two tables — openings and applications — plus a semantic-match function.
Files	Private object storage for résumés; the talent bank uses the bucket itself as its index, with a JSON manifest for AI annotations.
AI	Google Gemini — multimodal document reading, structured JSON generation, and 768-dimension embeddings.
Email	Gmail hand-off for candidate replies; RFC-compliant .ics generation for interviews.
Auth	Session-gated admin. Résumé access is authorised per request, never by obscure URL.
Schema cost	Interview scorecards, email history and cached assessments live in a JSON column — new HR capability ships without a migration.
Failure model	Parsing runs in the background, so a slow or failed AI call never blocks a candidate's submission. Failures are flagged and retryable from the admin.

Where this goes next.

NOT YET BUILT Everything in this section is planned, not shipped. Sections 01–07 describe the live product.

Candidate experience

NEAR TERM

- + Self-serve status page — a candidate checks where they stand without emailing you
- + Automatic acknowledgement on apply, with a realistic timeline
- + Withdraw-my-application and delete-my-data self-service (a DPDP data-principal right)
- + Availability picker so candidates choose an interview slot instead of trading emails

Recruiter throughput

NEAR TERM

- + Bulk actions — select many, pass on many, with one AI-drafted mail per person
- + Duplicate detection across the bank and applications, merged into one candidate view
- + Saved searches and role-based candidate alerts
- + Keyboard-first triage: score, decide, advance without leaving the queue

Interviewing

MID TERM

- + Multi-panel scorecards — several interviewers score independently, then compare
- + Structured interview kits generated per role, reused across candidates
- + Take-home assignments: issue, receive, and AI-assist the first review pass
- + Two-way calendar sync so booked slots respect real availability

Sourcing

MID TERM

- + Talent-pool re-engagement — when a role opens, surface past applicants who now fit
- + Referral tracking with attribution to the referrer
- + Job-board syndication from the same opening record
- + Careers-page analytics: views, starts, completions, drop-off per role

Offer & close

MID TERM

- + Offer letter generation from the role record, with an internal approval step
- + E-signature hand-off and acceptance tracking
- + Compensation bands per role, with an out-of-band warning before an offer goes out
- + Time-to-hire and offer-acceptance reporting

Fairness & audit

BIGGER BET

- + Blind screening mode — hide name, gender markers and institution during first pass
- + Score-drift monitoring across roles and reviewers
- + Full decision audit trail: who changed what, when, and on what evidence
- + Exportable compliance pack for a DPDP or client audit

Beyond hiring

BIGGER BET

- + Onboarding checklists that start the moment an offer is accepted
- + Employee records, documents and leave — hiring's natural continuation
- + WhatsApp as a candidate channel, which is where Indian candidates actually reply
- + Slack and WhatsApp alerts for the work queue, so urgency reaches you off-screen

Scale

BIGGER BET

- + Multi-tenant deployment, so this ships to clients as a product
- + Team roles and permissions — hiring manager vs recruiter vs interviewer
- + Regional and multi-language résumé handling
- + Public API and webhooks for downstream HRIS and payroll systems

Built by Voltair Tech.

An AI build studio in Mumbai. We ship systems like this one — designed around the decision a person actually has to make, not the data a database happens to hold.

voltairtech.com →