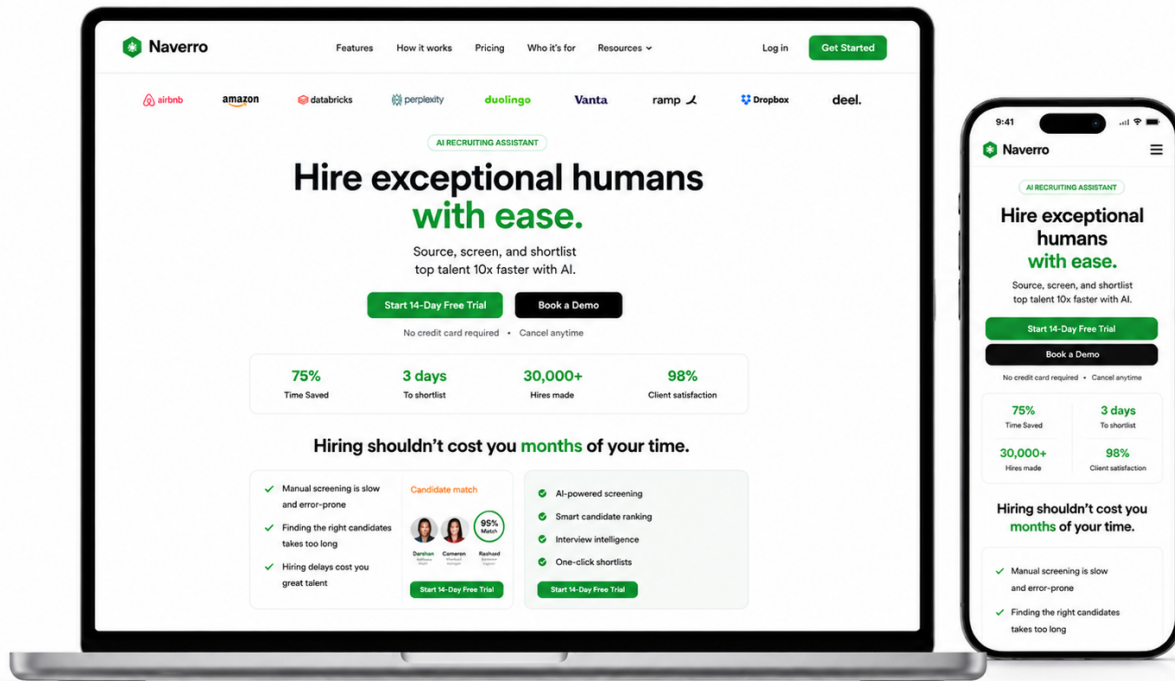


Naverro

An AI-driven hiring platform that turns a role brief into a scored, ranked shortlist, generating the job description and assessments, screening every applicant with AI, and carrying candidates through review, interviews, and hire in one auditable pipeline.



Tech Stack

Next.js • React • TypeScript • Tailwind CSS • Redux Toolkit • Prisma • PostgreSQL • Supabase • Python • FastAPI • RabbitMQ • OpenAI • Gemini • Grok • OpenRouter • Exa • AssemblyAI • Recall.ai • Cal.com • Stripe • SendGrid • Docker • Terraform • GitHub Actions • Google Cloud Run • Sentry

The Challenge

Hiring teams were overwhelmed by application volume. A single role could attract hundreds of candidates, forcing recruiters to spend countless hours reviewing CVs, conducting repetitive screening calls, and manually comparing applicants. By the time a shortlist was ready, top candidates had often accepted offers elsewhere.

Existing hiring tools failed to solve the problem. Applicant tracking systems relied on keyword matching, overlooking qualified candidates while rewarding those who optimized their resumes. Assessment platforms operated in isolated silos, making candidate comparisons inconsistent, and interview evaluations varied significantly between hiring managers with little transparency or evidence to support decisions.

Simply adding an LLM wasn't enough. Hiring decisions must be explainable, consistent, and defensible. Assessments need to be generated for each role, candidate identity must be verified during testing, and AI usage costs must remain predictable. What was needed was a unified platform that could create assessments, evaluate candidates consistently, explain every score, and remain cost-effective at scale.

Target Users

- In-house recruiters and talent acquisition teams
- Hiring managers and technical interviewers
- Startups and scale-ups hiring at volume
- Recruitment agencies and RPO providers
- Operations and finance owners tracking cost per hire

Solution

Naverio automates the entire hiring process, from defining a role to extending an offer. Recruiters describe the position, and the platform generates the job description, identifies and weights the required skills, and creates tailored assessments, including knock-out questions, role-specific skill tests, and voice or written screening questions. Candidates apply through a public careers page or are sourced in bulk, then complete the assessments online.

Each application is evaluated by AI that analyzes the CV, scores it against a weighted rubric, grades screening responses, and compares answers with the CV to generate a trust score. The platform combines these results into a ranked candidate profile, allowing recruiters to review the supporting evidence, move candidates through configurable hiring stages, schedule interviews, and monitor AI usage costs for every hire in real time.

Key Features

AI Role Setup: Turns a short brief into a complete job description, a weighted skill rubric, and a full assessment plan.

Knock-Out Questions: Generates hard-requirement questions that filter unqualified applicants before any AI spends.

CV Insights & Scoring: Parses résumés into structured data and scores each skill against the role's weighted rubric with a confidence level.

Verified Skill Tests: Builds role-specific multiple-choice assessments per candidate, with an automated verification pass that regenerates weak or duplicate questions.

Smart Screening: Replaces the first screening call with recorded voice or written answers, transcribed and graded against the same rubric.

Trust Score: Cross-checks a candidate's answers against their CV to surface inconsistencies before an interview is booked.

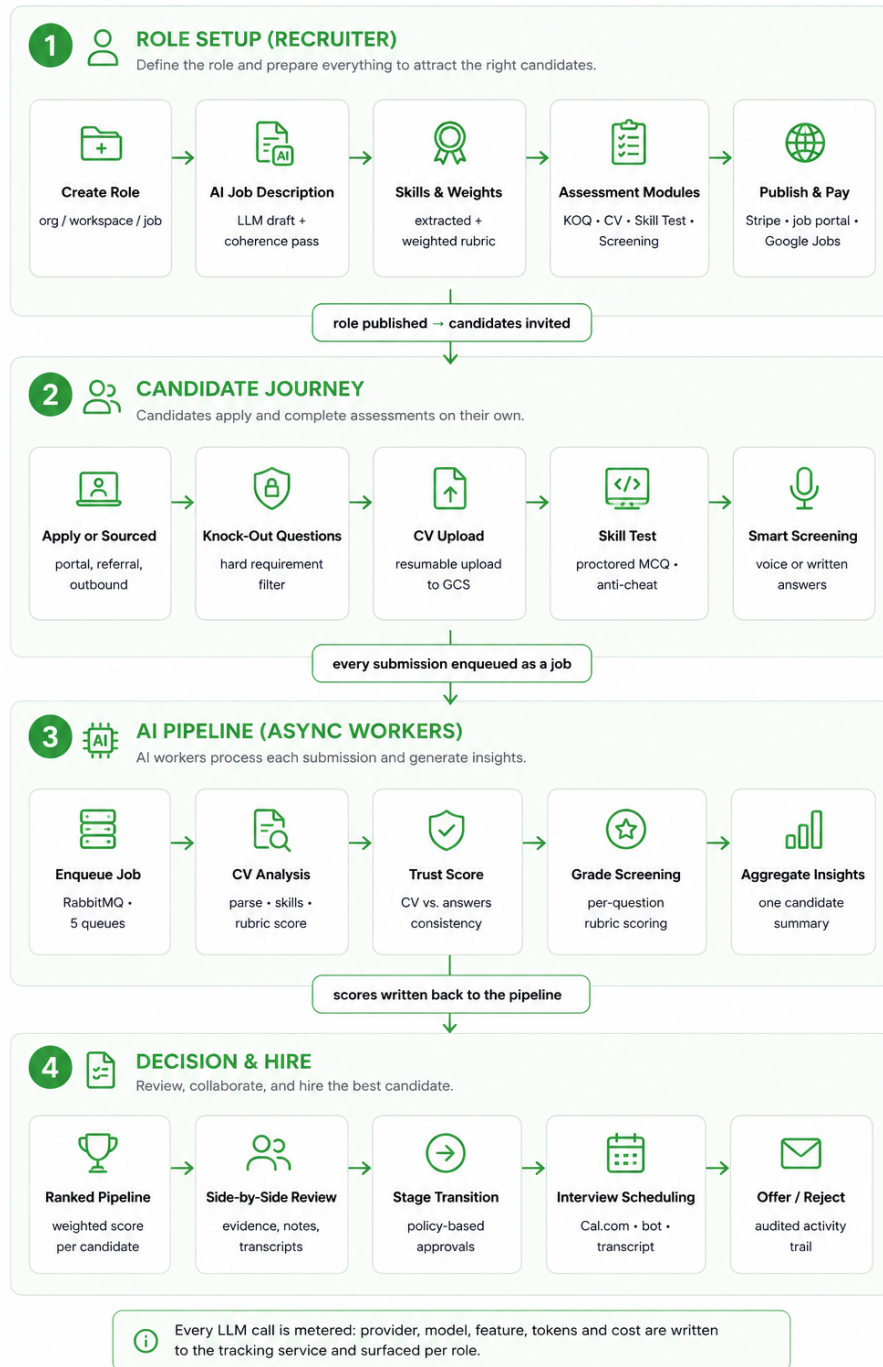
Anti-Cheat Proctoring: Configurable screenshot and camera capture, tab-blur detection, and key logging during assessments.

Ranked Pipeline & Review: One weighted score per candidate with the underlying evidence, scores, transcripts, notes, and the full activity trail — side by side.

Interview Scheduling: Collects interviewer availability, books rounds through Cal.com, and captures interview transcripts automatically.

LLM Cost Analytics: Meters every AI call by provider, model, feature, and role, so spend and margin per hire are always visible.

Workflow Diagram



Technical Architecture

Navero is built as three independently deployed services on Google Cloud Run: a Next.js frontend, an AI processing service, and a cost-tracking service. RabbitMQ acts as the message broker between them, ensuring long-running AI tasks are processed asynchronously without affecting the responsiveness of the user experience.



Challenges & Solutions

Challenge 1: High Application Volume

Challenge: Recruiters couldn't review, score, and compare hundreds of applications quickly without sacrificing quality.

Solution: Built an asynchronous processing pipeline using RabbitMQ and AI workers to analyze CVs, grade assessments, and aggregate candidate scores in parallel while keeping the application responsive.

Challenge 2: Keyword-Based Screening Missed Strong Candidates

Challenge: Traditional keyword matching overlooked qualified candidates and couldn't measure the importance of different skills.

Solution: Extracted and weighted skills from each job description, then evaluated CVs against a structured scoring rubric with confidence scores based on profile completeness.

Challenge 3: Role-Specific Assessments

Challenge: Static question banks quickly became repetitive and failed to evaluate role-specific skills accurately.

Solution: Built an AI-driven assessment generation pipeline with automated validation to detect duplicate, ambiguous, or incorrect questions before publishing.

Challenge 4: Candidate Trust & Assessment Integrity

Challenge: Remote assessments made it difficult to verify candidate authenticity and detect inconsistencies.

Solution: Implemented configurable proctoring with screenshot monitoring, tab-switch detection, and an AI-powered trust score that cross-checks assessment responses against the candidate's CV.

Challenge 5: Consistent Candidate Scoring

Challenge: Different parts of the platform produced inconsistent candidate scores due to varying aggregation logic.

Solution: Centralized scoring into a single shared engine with standardized rules, ensuring consistent results across the application, exports, and reporting.

Challenge 6: Tracking AI Costs

Challenge: AI costs were difficult to measure across multiple providers, models, and platform features.

Solution: Built a dedicated cost-tracking service that records every LLM request and aggregates usage by role, candidate, feature, and model using efficient SQL-based summarization.

Challenge 7: Flexible AI Provider Selection

Challenge: Switching AI providers or models required code changes and deployments.

Solution: Implemented configuration-driven provider selection using Secret Manager, enabling hot-swappable models, quality tiers, and automatic failover without redeployment.

Challenge 8: Multi-Tenant Data Isolation

Challenge: Multiple organizations needed to share the same platform while keeping their hiring data completely isolated.

Solution: Designed a multi-tenant permission system with organization, workspace, and job-level access controls, ensuring all data access is securely enforced on the server.

Production & Scalability

The platform is built with production-grade engineering practices: three containerised services, asynchronous processing behind a durable message broker, infrastructure as code, and separate staging and production Google Cloud projects so releases can be validated before they reach customers.

Containerised Architecture

Each service is built as an independent container image and deployed separately, so they scale and fail in isolation.

- **navero-web**: Next.js application: recruiter dashboard, candidate portal, public job portal, and all route handlers.
- **navero-llm (Artemis API)**: FastAPI service for synchronous AI generation during role setup.
- **navero-llm workers**: Worker manager running one supervised process per queue for long-running AI jobs.
- **navero-llm-tracking**: FastAPI service that records and aggregates LLM token usage and cost.
- **RabbitMQ**: Message broker with environment-tagged queues shared across all environments.

Authentication & Authorisation

Authentication runs on Supabase Auth with middleware-enforced sessions, email OTP, and Google sign-in. Authorisation is a three-tier role-based model: organisation, workspace, and job, resolved server-side into a permission checker, with a separate super-admin path for platform operations and per-organisation feature access tied to the billing plan.

Background Processing

Anything slow or expensive runs asynchronously on the worker fleet rather than inside a request. Queued jobs include:

- CV parsing, skill extraction, tagging, and rubric scoring
- Skill-test generation, verification, and question regeneration
- Smart screening transcription and rubric grading
- Trust score calculation across CV and candidate answers
- Candidate aggregation into a single ranked profile
- Scheduled maintenance, service-health evaluation, and alerting via Cloud Tasks

Resilience

The queue layer is built to survive broker and provider failures: a shared connection pool with robust reconnection, a circuit breaker that fails fast and self-heals through a half-open probe, per-job timeouts, status tracking with requeue on failure, and a graceful-drain shutdown sequence that lets in-flight jobs finish before a worker exits. LLM providers fall back to a secondary route when a primary key or model is unavailable, and generation retries with backoff.

Progress & Status Updates

Long-running AI work reports progress back to the UI through job status records and debounced cache invalidation in RTK Query, so recruiters watch CV analysis, test generation, and screening scores land per candidate without reloading — and without exposing database access to the browser.

Security

Production hardening covers both the platform and candidate data:

- Session-based authentication with server-side route protection and middleware
- Three-tier role-based access control with organisation-level data isolation
- Secrets held in Google Secret Manager and injected at runtime, never committed
- Service-to-service calls authenticated with signed service-account identity tokens
- Explicit CORS allow-lists per environment and internal API keys on webhook routes
- Input validation with Zod, HTML sanitisation, and ORM-parameterised queries
- Request rate limiting and signature verification on third-party webhooks

Database & Performance

The platform is engineered to handle a large relational dataset under sustained concurrent load while keeping candidate search and pipeline operations responsive.

- PostgreSQL (Supabase) with pgBouncer connection pooling for efficient connection management.
- 104-model relational schema maintained through 210+ version-controlled migrations.
- Fully normalized candidate data (experience, education, skills, projects, certifications) enabling fast, structured querying instead of document parsing.
- Pre-computed scoring and pipeline state columns to eliminate expensive recalculations during reads.
- Database-side aggregations and batched processing for analytics, minimizing memory usage and reducing application workload.
- Incremental processing that analyzes only newly added records, avoiding full dataset rescans.
- Resumable candidate uploads using browser workers and signed Cloud Storage URLs for reliable large-file transfers.

- Incrementally regenerated static job pages so the public careers portal serves cached HTML with minimal backend overhead.

Scalability

Each tier scales independently as demand grows.

Component	Scaling Strategy
Frontend	Stateless Cloud Run service, autoscaled per request volume
Artemis API	Additional Cloud Run instances behind the platform load balancer
AI Workers	More worker instances and per-queue concurrency for higher throughput
RabbitMQ	Durable queues with prefetch limits and pooled connections
PostgreSQL	Managed Postgres with pooled connections, backups, and replicas
Storage	Google Cloud Storage buckets per artefact type and environment
Cost Tracking	Stateless service scaled separately from the request path

Deployment

Deployment is fully automated through GitHub Actions. Each service is built into a container, pushed to Google Artifact Registry, and deployed to Cloud Run. Staging and production run in separate GCP projects, while infrastructure is managed with Terraform. Releases are versioned automatically, CI runs tests and code quality checks before every deployment, and Sentry monitors errors and performance in production.

My Responsibilities

I worked across both the Next.js application and the Python AI service, contributing around **430 commits** to the web platform and **45** to the AI backend. My focus was building recruiter-facing features, AI workflows, and the infrastructure around them.

- Built and maintained API endpoints for candidates, scoring, billing, and administration.
- Developed the recruiter dashboard, including candidate pipelines, filters, bulk actions, notes, and review workflows.
- Built the assessment configuration system for knockout questions, skill tests, screening logic, and AI service weighting.
- Developed the Python candidate aggregation pipeline, covering prompt engineering, queue processing, and output validation.
- Implemented the interview workflow, including scheduling, Cal.com integration, meeting links, scoring, and transcripts.
- Integrated Stripe subscriptions, credits, checkout, webhooks, and plan-based feature access.
- Built internal analytics and monitoring dashboards for AI services and platform health.
- Improved authentication with Google Sign-In, invitations, role-based access control, and approval workflows.
- Integrated Sentry across both services and optimized tracing to balance visibility with cost.
- Set up automated releases and resolved CI/CD deployment and secret management issues.
- Improved SEO for the public job portal with Google Jobs schema and incremental page regeneration.

Gallery

Recruiter Dashboard

NAVERO

Dashboard

Jobs

Outbound

Insights

Settings

Northwind Labs

Engineering

24 credits

FT

Jobs - Northwind Labs / Engineering

Senior Backend Engineer

Remote - Europe - Published 09 Mar 2026 - Standard plan

Export

Bulk message

Analyse 3 CVs

APPLICANTS

148

32 this week

PASSED KNOCK-OUT

61

41%

87 filtered automatically

FULLY ASSESSED

44

CV - skill test - screening

SHORTLISTED

9

Aug score 78%

Candidates

Pipeline

Interview rounds

Scoring rubric

Activity

Modules

Candidates

148 total - sorted by weighted average

Stage: all

Knockout: pass

Score >= 45

Search

WA SCORE	NAME	DATES	LOCATION	KNOCKOUT	CV SCORE	CV SKILL MATCH	SKILL TEST	SMART SCREENING	SUMMARY	FIT	STAGE	NOTES	COMMS	SOURCE
☐	92% Sarah Chen sarah.chen@mailnesia.com	Applied: 12 Mar Updated: 10 Mar	✓ Berlin, DE IP - match	High ■■■■■	Exceptional ■■■■■	■■■■■	Exceptional ■■■■■	Very Good ■■■■■	Strong distributed-systems depth; led Kafka migration	Strong ▾	Interview - R2	📄	💬	Portal
☐	87% Priya Patel priya.patel@mailnesia.com	Applied: 11 Mar Updated: 10 Mar	✓ London, UK IP - match	High ■■■■■	Very Good ■■■■■	■■■■■	Very Good ■■■■■	Exceptional ■■■■■	Excellent API design; Python + Go, strong on testing	Strong ▾	Interview - R1	📄	💬	Portal
☐	81% Ahmed Hassan ahmed.hassan@mailnesia.com	Applied: 12 Mar Updated: 10 Mar	✓ Cairo, EG IP - match	High ■■■■■	Very Good ■■■■■	■■■■■	Very Good ■■■■■	Very Good ■■■■■	Solid backend fundamentals; less exposure to k8s	Good ▾	Screening	📄	💬	Portal
☐	74% Michael Rodriguez michael.rodriguez@mailnesia.com	Applied: 10 Mar Updated: 17 Mar	✓ Austin, US IP - match	High ■■■■■	Very Good ■■■■■	■■■■■	Competitive ■■■■■	Competitive ■■■■■	Good service ownership; scaling experience is thinner	Good ▾	Screening	📄	💬	Portal
☐	71% Maya Yamamoto maya.yamamoto@mailnesia.com	Applied: 13 Mar Updated: 17 Mar	✓ Osaka, JP IP - match	Medium ■■■■■	Competitive ■■■■■	■■■■■	Competitive ■■■■■	Competitive ■■■■■	Strong data modeling; limited Go experience	Maybe ▾	Skill test	📄	💬	Portal
☐	66% James Kim james.kim@mailnesia.com	Applied: 09 Mar Updated: 10 Mar	✓ Seoul, KR IP - match	Medium ■■■■■	Competitive ■■■■■	■■■■■	Competitive ■■■■■	Competitive ■■■■■	Competent; mostly non-stick; Rails background	Maybe ▾	Skill test	📄	💬	Portal
☐	58% Emily Thompson emily.thompson@mailnesia.com	Applied: 11 Mar Updated: 10 Mar	✓ Dublin, IE IP - match	Medium ■■■■■	Competitive ■■■■■	■■■■■	Needs Improvement ■■■■■	Needs Improvement ■■■■■	Junior-leaning for the seniority band	Maybe ▾	CV review	📄	💬	Portal
☐	49% Raj Kumar raj.kumar@mailnesia.com	Applied: 10 Mar Updated: 15 Mar	✓ Bengaluru, IN IP - match	Low ■■■■■	Needs Improvement ■■■■■	■■■■■	Needs Improvement ■■■■■	Needs Improvement ■■■■■	Requirement gaps on ownership and system design	Weak ▾	CV review	📄	💬	Portal
☐	— Isabella Garcia isabella.garcia@mailnesia.com	Applied: 14 Mar Updated: 14 Mar	✓ Madrid, ES IP - match	Not started ■■■■■	Queued ■■■■■	—	—	—	CV analysis queued	— ▾	Applied	📄	💬	Portal

Role Setup & Assessment Modules

NAVERO

Dashboard

Jobs

Outbound

Insights

Settings

Northwind Labs · Engineering

24 credits

FT

Jobs · Senior Backend Engineer · Setup

Configure assessment modules

Each module is generated from the job description and its weighted skill rubric.

Save draftPublish role

Setup progress

3 / 7

Job description

generated

Skills & weights

14 skills

Knock-out questions

6

Skill test

generating

Smart screening

VS

Proctoring level

Service weights

Review & publish

Knock-Out Questions

BINARY

Saved

Check for must-have skills, years experience, certificates, location or language requirements

6 questions

Configure

CV Insights

CV

Saved

Evaluate candidate skills and experience against a consistent, unbiased scoring rubric

14 skills weighted

Configure

Skill Test

NCO

In progress

Assesses candidates' real-world knowledge through short skill tests

5 topics · 24 questions

Configure

Smart Screening

VS

Not started

Analyse in-depth written or spoken responses to questions specific to the role

4 questions · voice

Configure

Talent Sourcing

SOURCING

Not started

Source top talent via our pool of 200m+ candidates from 130+ data sources

Not configured

Configure

Weighted skill rubric

auto-extracted

REQUIRED

Go

Advanced · w5

Distributed systems

Advanced · w5

PostgreSQL

Intermediate · w4

Kubernetes

Intermediate · w3

PREFERRED

gRPC

Intermediate · w2

Terraform

Beginner · w2

CV Insights

NAVERO

Dashboard

Jobs

Outbound

Insights

Settings

Northwind Labs · Engineering

24 credits

FT

CV Analysis Results — Sarah Chen

Analysis ✓

confidence: high

Reanalyse

View PDF

Sarah Chen

sarah.chen@mailnesia.com

8 yrs experience

4 roles · 2 promotions

Berlin, Germany

matches role location ✓

MSc Computer Science

TU Munich · 2017

Skill scores vs. rubric

Go · Required · needs Advanced

✓ 4.6 / 5

Distributed systems · Required · needs Advanced

✓ 4.4 / 5

PostgreSQL · Required · needs Intermediate

✓ 3.8 / 5

Kubernetes · Required · needs Intermediate

— 2.9 / 5

gRPC · Preferred · needs Intermediate

— 2.4 / 5

Terraform · Preferred · needs Beginner

— 1.2 / 5

MANAGER SUMMARY

Eight years of backend engineering with clear ownership of high-throughput services. Led a Kafka migration handling ~40k events/sec and reduced p99 latency by 38%. Go and distributed systems are evidenced across two roles. Kubernetes exposure is real but operational rather than architectural, and Terraform appears only once in passing — the main gap against the rubric.

WEIGHTED TOTAL

94%

Exceptional

Interview: yes

confidence 0.86

Extracted work history

Staff Backend Engineer

Zalando · 2022 – present

Senior Backend Engineer

Delivery Hero · 2019 – 2022

Backend Engineer

SoundCloud · 2017 – 2019

Organisation CV tags

event-driven

high-scale

team lead

EU work permit

Go

Kafka

Skill Test Builder

NAVERO

DashboardJobsOutboundInsightsSettings

Northwind Labs · Engineering

24 credits

FT

Jobs · Senior Backend Engineer · Skill Test

Skill test

24 questions across 5 topics, generated for this role and verified automatically.

+ Add topic

Regenerate

Approve test

Topics

24 questions

Concurrency in Go

Advanced · 6 questions

Verified

Distributed consensus

Advanced · 5 questions

Verified

PostgreSQL query tuning

Intermediate · 6 questions

Verified

Kubernetes operations

Intermediate · 4 questions

1 regenerated

gRPC & API contracts

Intermediate · 4 questions

Verifying...

Verification pass

Generated24

Flagged ambiguous2

Flagged duplicate1

Wrong answer key1

Regenerated with context4

Flagged questions are regenerated with the already-approved set passed in as context, so replacements never repeat an existing question.

Concurrency in Go · question 3 of 6

Advanced

Verified

A worker pool reads from an unbuffered channel and writes results to a second unbuffered channel that no consumer is reading yet. The pool is cancelled via context. What is the most likely outcome?

☐ All workers return immediately because the context cancellation propagates to the channel operations.

☒ Workers block on the result send and leak until the send is also selected against ctx.Done().

☐ The runtime panics with "all goroutines are asleep — deadlock".

☐ The results channel buffers the pending sends until a consumer attaches.

RATIONALE

Cancelling a context does not unblock a channel send. Unless the send is wrapped in a select with ctx.Done(), each worker stays parked and the goroutine leaks — the failure mode candidates most often miss.

skill: Go

discrimination: high

distractors: plausible

Smart Screening Review

NAVERO

DashboardJobsOutboundInsightsSettings

Northwind Labs · Engineering

24 credits

FT

Jobs · Senior Backend Engineer · Candidates · Sarah Chen

Smart screening review — Sarah Chen

3 voice answers · transcribed automatically · graded against the role rubric

Aggregate 89%

Export

Advance stage

Q1

Advanced

voice · 1:47

Walk me through a service you owned end to end — how did you decide its boundaries?

TRANSCRIPT

I owned the order-events service at Zalando. We drew the boundary around the lifecycle of an order event rather than around the team, because the previous split had two teams writing to the same table. I started from the consumers — fulfillment, notifications, analytics — and worked backwards to a single owner for the event contract, then versioned it so consumers could migrate independently...

Role-specific88%

Core skills84%

weighted 60 / 40 for this skill level

Q2

Advanced

voice · 1:54

Describe a time your change caused a production incident. What did you do?

TRANSCRIPT

We shipped a Kafka consumer-group rebalance config that looked harmless in staging. In production the rebalance storm stalled the fulfillment pipeline for about nine minutes. I rolled back first, then wrote the postmortem — the real cause was that staging had one partition per topic, so rebalance behaviour was never exercised. We changed staging to mirror partition counts...

Role-specific91%

Core skills86%

weighted 60 / 40 for this skill level

Q3

Intermediate

voice · 1:61

How do you decide between adding an index and rewriting a query?

TRANSCRIPT

I look at the plan first. If the planner is already choosing a sequential scan because the predicate isn't selective, an index won't help and I'd rewrite. If it's a nested loop with a high row estimate error, I'd check statistics before touching either...

Role-specific74%

Core skills79%

weighted 60 / 40 for this skill level

Trust Score & Knock-Out Results

NAVERO

Dashboard

Jobs

Outbound

Insights

Settings

Northwind Labs

Engineering

24 credits

FT

Knock-out results — Sarah Chen

sarah.chen@mailnesia.com

Trust: High

Close

Required questions

5 of 6 met

QUESTION	TYPE	ANSWER	REQUIREMENT	RESULT	GATE
Do you have the right to work in the EU without sponsorship?	boolean	Yes	Yes	Pass	Required
How many years of professional Go experience do you have?	number	7	min 5	Pass	Required
Have you owned a service in production with on-call responsibility?	boolean	Yes	Yes	Pass	Required
Are you able to work within CET ± 2 hours?	boolean	Yes	Yes	Pass	Required
What is your expected annual base salary (EUR)?	compensation	€95,000	max €110,000	Pass	Required
Do you hold a CKA certification?	boolean	No	Yes	Fail	Optional

TRUST SCORE

88

High

Answers are consistent with the CV. No contradictions detected.

Consistency checks

✓

Years of Go experience

claimed 7 - CV supports 7

✓

Production ownership

on-call evidenced at 2 employers

✓

Location & work rights

Berlin - EU permit on CV

✓

CKA certification

answered no - none on CV

!

Kubernetes depth

self-rated high - CV shows operational use only

PROMPT-INJECTION SCAN

Clean

no instruction text found in CV

Interview Scheduling

NAVERO

Dashboard

Jobs

Outbound

Insights

Settings

Northwind Labs

Engineering

24 credits

FT

Jobs - Senior Backend Engineer - Interview rounds

Interview rounds

3 rounds configured - availability collected from 5 interviewers - booked through Cal.com

+ Add round

Chase availability

Send invites

R1

Technical deep dive

60 min - Google Meet

2 interviewers

Daniel Lee - Staff Engineer

Zara Khan - Eng Manager

note-taking bot on

PREP KIT QUESTIONS

1

Walk me through the hardest scaling problem you have owned. What did you measure, and what did you change?

Role Capability

probes - 3

2

Tell me about a decision you made with incomplete information that turned out to be wrong.

Problem Solving & Judgment

probes - 3

3

How do you get a reluctant team to adopt a standard you believe in?

Communication & Collaboration

probes - 3

R2

System design

75 min

awaiting availability

Lucas Silva

Sophia Martinez

1 of 2 responded

R3

Hiring manager & values

45 min

locked until R2 complete

Upcoming bookings

4

Sarah Chen

Thu 19 Mar - 10:00 CET - R2 System design

Confirmed

Priya Patel

Thu 19 Mar - 14:00 CET - R1 Technical

Confirmed

Ahmed Hassan

awaiting candidate - R1 Technical

Invited

Michael Rodriguez

Fri 20 Mar - 16:00 CET - R1 Technical

Rescheduled

Interviewer availability

Daniel Lee

12 slots

Zara Khan

8 slots

Lucas Silva

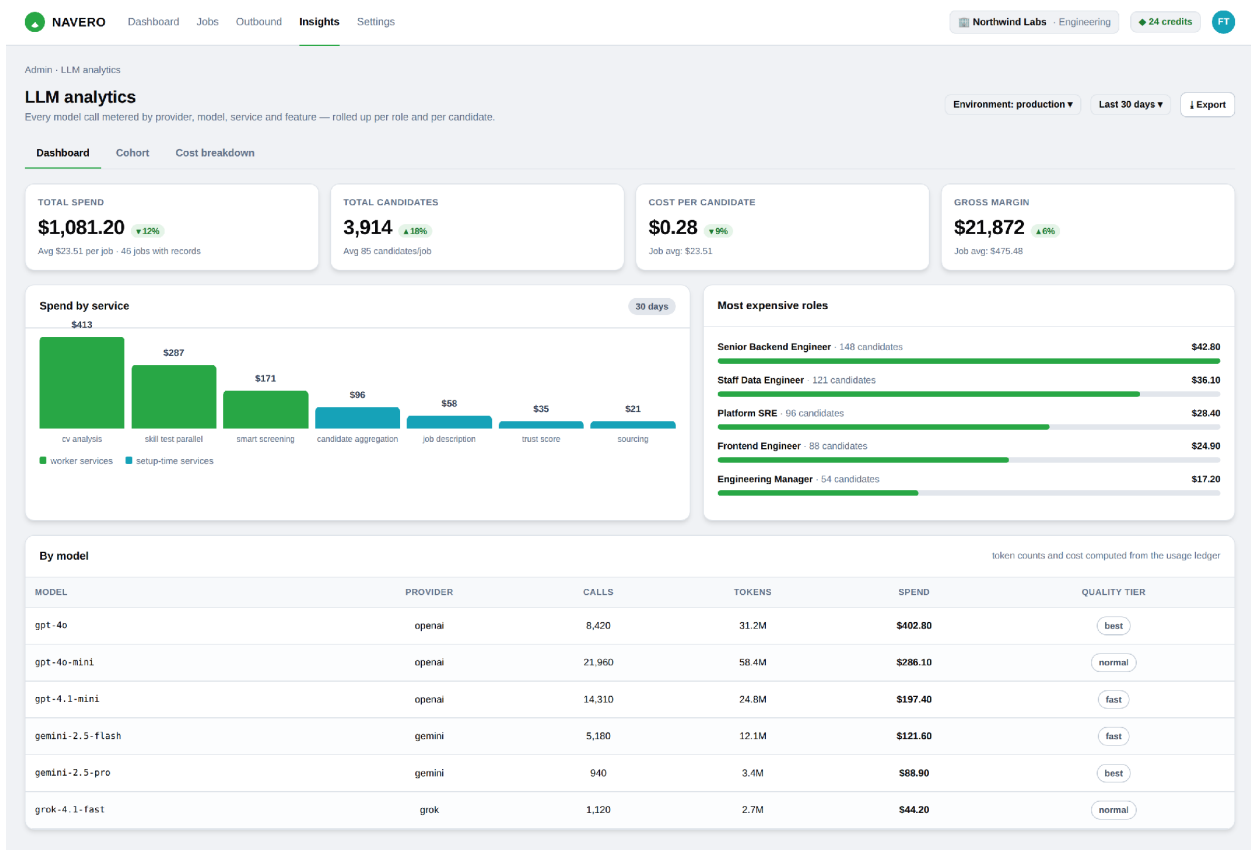
5 slots

Sophia Martinez

no response

Chase reminders send automatically every 24 h.

LLM Cost Analytics



Project Links

Live Link: www.navero.me

Github repos: Can't share because of client privacy.

Final Takeaway

Navero demonstrates that building an AI hiring platform is about far more than integrating large language models. The real challenge was creating a system that produces consistent, explainable, and trustworthy hiring decisions. From role-specific assessment generation and standardized scoring to candidate verification, configurable AI providers, and real-time cost tracking, every part of the platform was designed for transparency and reliability. The result is a production-ready hiring platform that automates the journey from job creation to ranked shortlist, while ensuring every recommendation can be traced back to the evidence behind it.