

AI JOB SEARCH AGENT

ESSENTIAL PRODUCT DOCUMENTS

From natural-language intent to explainable job matching and application tracking

PRODUCT LIFECYCLE PORTFOLIO EDITION

PORTFOLIO ROLE	PRODUCT STAGE	EVIDENCE	VALIDATION
AI PM	MVP	ACTUAL	PENDING
Product strategy, AI design and delivery	Functional technical-feasibility build	Repository-backed implementation	No production user outcomes claimed

Prepared by Min Thu Kyaw

AI Product Manager | July 2026 | Version 1.0

github.com/minthukyawdev-droid/ai-job-search-agent

Document Control

Publication status, evidence conventions, portfolio scope and revision history

Field	Value
Document	AI Job Search Agent - Essential Product Documents
Edition	Final Portfolio Edition
Document code	AJSA-PORT-01
Version	1.0
Status	Functional MVP; controlled pilot not yet completed
Author	Min Thu Kyaw
Role	AI Product Manager
Primary audience	Hiring managers, product leaders, AI leaders, engineering leaders and portfolio reviewers
Source of truth	Public GitHub repository and deployed build status
Last updated	July 2026

Evidence convention

This portfolio separates what exists from what still requires validation. The distinction prevents technical implementation, planning targets, and hypothetical business outcomes from being presented as the same type of evidence.

Label	Meaning
Actual	Implemented behavior, repository evidence, configuration, deployment status or completed artifact.
Assumption	A belief about users, market behavior, feasibility or viability that requires validation.
Target	A proposed success threshold; not an achieved result.
Recommendation	A product decision based on current evidence and identified risk.
Deferred	Intentionally excluded from the MVP or blocked by a later phase gate.

PORTFOLIO INTEGRITY RULE

No user count, retention rate, interview conversion, revenue result or recommendation-quality score is reported as achieved. The MVP demonstrates technical feasibility; customer value and commercial viability remain to be validated.

Essential artifact set

Phase	Essential artifact	Purpose	Status
1 - Conceive	Product Concept Document (PCD)	Define the customer problem, product thesis, target users, value proposition and critical assumptions.	Portfolio complete
2 - Plan	Product Requirements Document (PRD)	Lock the MVP boundary, prioritize capabilities and translate the concept into testable requirements.	Portfolio complete
2 - Plan	AI Strategy & Evaluation Plan	Define AI responsibility, ranking logic, fallbacks, quality metrics and trust boundaries.	Portfolio complete
3 - Develop	As-Built Product & Architecture Report	Document what was actually implemented, trace requirements and identify release gaps.	Repository-backed
4 - Launch	Pilot Measurement & Roadmap Plan	Define launch cohort, analytics, experiments, gates and	Proposed

Phase	Essential artifact	Purpose	Status
		evidence needed for scale.	

Revision history

Version	Date	Owner	Change
0.1	July 2026	Min Thu Kyaw	Initial repository review and full narrative case study.
1.0	July 2026	Min Thu Kyaw	Restructured into a concise lifecycle portfolio using only essential product artifacts.

Contents

One consolidated portfolio document organized by product phase

01	Executive Portfolio Summary Product, role, implemented value and current recommendation
02	Phase 1 - Conceive / PCD Problem, target users, JTBD, product thesis, positioning, assumptions and concept gate
03	Phase 2 - Plan / PRD Goals, MVP, prioritization, user stories, requirements, non-functional requirements and release criteria
04	Phase 2 - Plan / AI Strategy AI boundary, hybrid ranking, explainability, fallback design, evaluation and responsible AI
05	Phase 3 - Develop / As-Built Architecture, data model, API surface, traceability, readiness and P0 backlog
06	Phase 4 - Launch / Measurement Pilot design, North Star, funnel, events, experiments, roadmap and scale gate
07	Product Manager Reflection Trade-offs, lessons, gaps and final recommendation
08	Appendix Repository evidence map and implementation references

RECOMMENDED REVIEW PATH

A hiring manager can read the Executive Summary, phase decisions, As-Built traceability and Product Manager Reflection in approximately 15 minutes. The detailed requirements and measurement sections provide deeper evidence for product and technical interviews.

Executive Portfolio Summary

The customer problem, product response, implementation evidence and current decision

CATEGORY	BEACHHEAD	CORE LOOP	CURRENT GATE
AI JOB SEARCH	TECH TALENT	SEARCH → ACT	ITERATE
Candidate decision-support workspace	Product, engineering, data and AI roles	Discover, explain, save, apply and track	Ready for controlled validation after P0 gaps

Executive overview

AI Job Search Agent is a full-stack job discovery and application-management product that allows a candidate to describe the work they want in natural language, receive ranked opportunities, understand why each job matches, and track the opportunity from saved to applied, interview or rejected.

The product is intentionally positioned as a candidate decision-support system rather than another undifferentiated job board. Its value comes from combining natural-language intent, structured filters, semantic search, deterministic matching signals, transparent explanations, candidate profile context and a lightweight application pipeline.

My contribution

Area	Product management contribution
Discovery and strategy	Defined the problem, candidate segments, Jobs To Be Done, value proposition, product principles and validation assumptions.
MVP planning	Converted the concept into a focused end-to-end loop and explicitly deferred high-risk automation.
AI product design	Separated query understanding, embedding retrieval, deterministic ranking and explanation generation into distinct responsibilities.
Requirements and delivery	Specified core workflows, data entities, acceptance criteria, dependencies, release gates and technical trade-offs.
Trust and measurement	Designed transparent score breakdowns, deterministic fallbacks, AI evaluation metrics, pilot analytics and guardrails.
Technical implementation	Built or directed a Next.js frontend, FastAPI backend, PostgreSQL persistence, Redis caching, provider ingestion and CI/CD setup.

Problem and implemented response

Candidate problem	Product response
Search requires repeated keywords and filters.	Natural-language search extracts role, skills, seniority, location and remote intent.
Relevant roles may use unfamiliar titles.	Embeddings compare semantic meaning rather than exact title matches alone.
Large result sets create review fatigue.	Hybrid ranking prioritizes opportunities using semantic, skill, keyword and location signals.
Match scores can feel opaque.	Every result includes an explanation and visible score breakdown.
Candidate context is lost between searches.	A reusable profile stores skills, experience, preferred roles, location and resume text.
Saved jobs and application progress become fragmented.	The product supports saved, applied, interview and rejected statuses.

Current product evidence

- **Actual** - FastAPI endpoints support job import, search, profile creation, recommendations, save/apply actions and application retrieval.
- **Actual** - The ranking model uses 50% semantic similarity, 20% skill match, 20% keyword match and 10% location match, with a seniority adjustment.
- **Actual** - Job adapters exist for Remotive, Adzuna, Greenhouse, Lever and controlled JSON imports.
- **Actual** - The user interface shows extracted filters, overall match score, provider, explanation and detailed signal breakdown.
- **Actual** - PostgreSQL and Redis are configured, and CI checks backend smoke behavior plus the production frontend build.
- **Not yet proven** - No real pilot, retention result, relevance benchmark, interview conversion or willingness-to-pay result is claimed.

CURRENT RECOMMENDATION - ITERATE

The MVP demonstrates the complete product loop and credible AI architecture. Before a public pilot, add authentication, profile update/delete controls, analytics instrumentation, job freshness handling, privacy safeguards for resume data, structured error UX and a versioned relevance benchmark.

PHASE 1 - CONCEIVE

Product Concept Document

Customer problem, product thesis, strategic wedge and evidence required before deeper investment

PHASE
DECISION

GO

1. Product Concept

Vision, mission, category, product principles and initial solution boundary

Product definition

AI Job Search Agent is an explainable, personalized job-search workspace for candidates who need to identify strong opportunities without manually reviewing large volumes of weak matches. It converts unstructured career intent into structured search signals, ranks normalized jobs, explains the result and supports the candidate's application workflow.

Vision	Mission
Help every candidate focus on the opportunities most likely to advance their career.	Turn career goals, experience and preferences into explainable job recommendations and an organized application workflow.

Product principles

Principle	Design consequence
Relevance over volume	Optimize for a smaller set of actionable opportunities rather than an endless feed.
AI interprets; evidence explains	Use AI for flexible understanding, but expose deterministic signals and source data.
Candidate control	The user decides what to save, apply to, edit or ignore; no silent autonomous applications.
Graceful degradation	Core search and explanation behavior must remain useful when the LLM is unavailable.
One workflow, not one feature	Connect search, recommendations, job details, saved jobs and application status.
Privacy by design	Resume and career data require consent, deletion controls and production-grade access protection.

Conceptual experience

Stage	User-visible outcome
Profile	The candidate teaches the system what good looks like using skills, experience, preferred roles, location and resume text.
Intent	The candidate describes the work they want in ordinary language.
Interpret	The product exposes extracted role, skills, seniority, location and remote intent.
Rank	The system orders normalized jobs by a hybrid fit score.
Explain	Each job shows why it matches and how each signal contributed.
Act	The candidate saves the role, opens the original source or marks it as applied.
Track	The candidate updates the opportunity through saved, applied, interview or rejected.

Product boundary

WHAT THE PRODUCT OWNS

The product owns candidate intent interpretation, relevance ranking, recommendation explanation and application-state organization. It does not own employer hiring decisions, guarantee interviews, or replace the original job provider's application flow.

2. Customer Problem & Opportunity

Target users, unmet needs, Jobs To Be Done and the strategic product wedge

Primary customer problem

Job seekers can access thousands of listings but still perform the most difficult work manually: converting career goals into search syntax, identifying equivalent titles, judging fit, comparing requirements and organizing follow-up. Search supply is abundant; decision confidence is scarce.

Target user segments

Segment	Context	Highest-value outcome
Focused professional	Knows the role family and wants fewer, stronger matches.	Spend attention only on relevant roles.
Career switcher	Has transferable experience but may not know the employer's job-title taxonomy.	Discover adjacent roles and understand transferable fit.
High-volume applicant	Searches across several sources and manages many applications.	Keep discovery and progress in one workspace.
Remote or regional candidate	Needs location, remote and regional eligibility to be interpreted correctly.	Avoid wasting time on roles that are not practically accessible.

Primary persona

Attribute	Representative portfolio persona
Role	Mid-to-senior technology professional seeking an AI, product, engineering or data role.
Behavior	Uses multiple job boards, saves links in different places and rewrites similar search queries.
Goals	Find high-fit jobs quickly, understand gaps, apply selectively and keep the pipeline organized.
Frustrations	Noisy results, inconsistent titles, unclear match logic, repeated filters and lost follow-up.
Success statement	"I can identify the five opportunities worth my time and understand why each one fits."
Evidence status	Illustrative persona; requires primary interviews and behavioral validation.

Jobs To Be Done

Job	JTBD statement
Discover	When I begin a search, help me express the role I want naturally so I do not need to understand every platform's filter taxonomy.
Evaluate	When I review an opportunity, help me understand the strongest fit and mismatch signals so I can decide whether it deserves attention.

Job	JTBD statement
Personalize	When I return, remember my background and preferences so I do not rebuild context from zero.
Organize	When I save or apply, keep the opportunity and status in one place so I know what to do next.
Trust	When AI recommends a job, show the evidence and limitations so I can make the final decision.

Opportunity statement

STRATEGIC WEDGE

Traditional job boards optimize inventory and filtering. AI Job Search Agent can differentiate by optimizing candidate attention: understanding intent, ranking opportunities across inconsistent titles and explaining why a result deserves action.

3. Product Strategy & Value Proposition

Positioning, value exchange, business hypothesis and competitive differentiation

Positioning statement

For technology professionals overwhelmed by noisy job feeds, AI Job Search Agent is an explainable career-opportunity workspace that converts natural-language goals and candidate context into ranked, transparent job matches and an organized application pipeline. Unlike filter-first job boards, it combines semantic understanding with visible fit signals and candidate-controlled action tracking.

Value proposition

Customer job or pain	Product value
Describe a complex role preference.	Natural-language intent replaces repeated filter setup.
Recognize equivalent roles and transferable skills.	Semantic matching identifies context beyond exact titles.
Decide which job deserves attention.	Hybrid score and explanation reduce manual comparison effort.
Build confidence in AI recommendations.	Extracted filters and score components remain visible.
Manage the search as an ongoing process.	Profile context, recommendations and status tracking persist.

Lean Canvas - concept hypothesis

Block	Portfolio hypothesis
Problem	Noisy search, inconsistent titles, manual fit assessment and fragmented application tracking.
Customer segment	Technology professionals, career switchers and high-intent active applicants.
Unique value proposition	Describe the work you want; see the roles worth your attention and why.
Solution	Intent parsing, hybrid ranking, explainability, profiles, recommendations and tracking.
Channels	Product-led web experience, portfolio/demo distribution, career communities and future partnerships.
Revenue hypothesis	Freemium search and tracking; premium advanced matching, alerts, resume tailoring and analytics.
Cost drivers	Job data access, LLM and embedding inference, hosting, observability, product development and support.
Key metrics	High-intent opportunities saved or applied to, relevance feedback, retained weekly usage and interview conversion.
Potential advantage	Candidate preference and interaction data can improve ranking and workflow personalization over time, subject to privacy controls.

Critical assumptions

ID	Assumption	Risk if false	Validation
A1	Candidates prefer natural-language intent for complex searches.	The core interaction adds little value.	Usability test versus filter-first search.
A2	Hybrid ranking produces more actionable results than keyword-only ranking.	Technical differentiation does not improve outcomes.	Offline benchmark plus search-to-save experiment.
A3	Users value explanations enough to inspect or act on them.	Explainability adds interface complexity without trust benefit.	Score-only versus explanation experiment.
A4	Profile context improves recommendation quality.	Profile setup creates friction without relevance gain.	Structured profile versus query-only evaluation.
A5	Search and tracking in one product improves repeat use.	Users continue using external spreadsheets or job boards.	Cohort retention and status-update behavior.
A6	A viable provider mix can deliver sufficient fresh job supply.	Recommendations become stale or narrow.	Provider quality, freshness and coverage audit.

Conceive phase exit decision

GO TO PLAN

The problem is meaningful, the product wedge is coherent and the technical approach is feasible enough for a bounded MVP. Proceed to detailed planning, but treat desirability and viability as unvalidated until candidate interviews and pilot behavior provide evidence.

PHASE 2 - PLAN

Product Requirements Document

MVP boundary, prioritized capabilities, requirements, acceptance criteria and release decision

PHASE
DECISION

GO

4. PRD Decision Rule

A feature belongs in the MVP only when it delivers or measures the complete intent-to-action loop

MVP RULE

Include a capability only when it is required to complete or evaluate this workflow: create profile → describe intent → inspect extracted filters → review ranked and explained jobs → save or apply → update application status.

Product goals

Goal type	Goal
User	Reduce the effort required to identify relevant jobs and organize next actions.
Product	Prove that natural-language intent, transparent hybrid ranking and workflow continuity create a useful experience.
AI	Deliver grounded, inspectable recommendations with deterministic fallbacks and measurable quality.
Technical	Establish a modular full-stack foundation that supports provider ingestion, persistence, caching and future evaluation.
Learning	Instrument the funnel and collect evidence required to decide whether to iterate, scale or stop.

Non-goals for the MVP

- Automatic or bulk job application submission.
- Guaranteeing interview or employment outcomes.
- Recruiter-facing applicant tracking or employer accounts.
- Real-time browser automation across job boards.
- Full resume rewriting, cover-letter generation or interview coaching.
- Native mobile applications.
- Paid subscriptions, billing and enterprise administration.
- Universal global job coverage or guaranteed provider availability.

5. MVP Scope & Prioritization

What must be delivered now, what can follow and why the scope is intentionally narrow

MVP capability map

Capability	Priority	Rationale	Evidence
Natural-language search	Must	Primary interaction and differentiation.	Implemented
Structured filter extraction	Must	Makes interpretation visible and testable.	Implemented

Capability	Priority	Rationale	Evidence
Hybrid relevance ranking	Must	Converts retrieved jobs into prioritized decisions.	Implemented
Explainable score breakdown	Must	Supports trust and ranking diagnosis.	Implemented
Candidate profile	Must	Creates persistent personalization context.	Implemented
Personalized recommendations	Must	Tests value beyond one-off search.	Implemented
Save and application tracking	Must	Completes the action workflow.	Implemented
Multi-provider ingestion	Should	Creates enough supply for meaningful testing.	Implemented with provider dependencies
Authentication and account controls	Must before pilot	Protects resume and career data.	Gap
Analytics instrumentation	Must before pilot	Required to validate the product thesis.	Gap
Freshness and expiry controls	Should before pilot	Prevents stale opportunities from damaging trust.	Gap
Automated applications	Won't in MVP	High trust, compliance and quality risk.	Deferred

Prioritization logic

Factor	Question	Application
User impact	Does this materially improve discovery, decision confidence or workflow continuity?	Ranking, explanations and tracking score high.
Learning value	Does this help validate the central product hypothesis?	Analytics and relevance feedback become P0 for pilot.
Risk reduction	Does this prevent privacy, trust or availability failure?	Authentication, deletion and provider health checks become release gates.
Dependency	Is the capability required by several later features?	Profiles and normalized job ingestion unlock recommendations.
Effort	Can the team deliver a credible version without creating a platform rebuild?	Advanced automation and enterprise controls remain deferred.

Release slices

Release	Objective	Included
MVP 0.1 - Feasibility	Prove the full technical loop.	Sample/imported jobs, search, profile, ranking, explanations, save/apply and basic deployment.
Pilot 0.2 - Validation	Measure relevance, trust and repeated use safely.	Authentication, analytics, feedback, freshness, privacy controls, observability and benchmark.
Beta 0.3 - Workflow	Improve repeated job-search management.	Alerts, notes, reminders, richer profile controls and application timeline.
Future 1.0 - Assistance	Support higher-value career actions.	Resume tailoring, interview preparation and carefully reviewed application assistance.

6. Core User Stories & Acceptance Criteria

Testable behavior for the highest-value product journeys

ID	Story	User story	Acceptance summary
US-01	Search by intent	As a candidate, I want to describe the role I want naturally so that I can search without learning a filter taxonomy.	A non-empty query returns extracted filters and ranked jobs; the filters are visible; errors are actionable.
US-02	Understand fit	As a candidate, I want to see why a job matches so that I can decide whether it deserves attention.	Every ranked result has a match score, explanation and score breakdown; unsupported claims are not introduced.
US-03	Create profile	As a candidate, I want to save my skills, experience, role and	The profile persists; a profile embedding is created; the user

ID	Story	User story	Acceptance summary
		location preferences so that recommendations reflect my background.	can later update or delete it before public pilot.
US-04	Receive recommendations	As a candidate, I want jobs ranked against my profile so that I can discover opportunities without repeating a query.	Recommendations use profile context and return the same explanation structure as search results.
US-05	Save and apply	As a candidate, I want to save or mark a job as applied so that I can act without losing the opportunity.	The action is tied to the authenticated user and job; duplicate records are prevented; state is immediately visible.
US-06	Track status	As a candidate, I want to update a job to interview or rejected so that I can manage my pipeline.	The status is restricted to allowed values, persists and appears in reverse chronological order.

7. Functional Requirements

The minimum testable capabilities required for the MVP and controlled pilot

ID	Module	Requirement	Status
FR-101	Profile	Create a candidate profile containing name, skills, experience, preferred roles, location preference and optional resume text.	Implemented
FR-102	Profile	Generate and persist a profile embedding from normalized profile context.	Implemented
FR-103	Profile	Allow an authenticated user to update, export and delete profile data.	Pilot gap
FR-201	Search	Accept a natural-language query of at least two characters.	Implemented
FR-202	Search	Extract role, skills, seniority, location and remote preference using AI or deterministic parsing.	Implemented
FR-203	Search	Show the extracted filters to the user.	Implemented
FR-204	Search	Retrieve and rank jobs using semantic, skill, keyword and location signals.	Implemented
FR-205	Search	Return a grounded explanation, matched skills and score breakdown.	Implemented
FR-301	Recommendations	Rank jobs using stored profile context and location preference.	Implemented
FR-302	Recommendations	Cache recommendation responses and invalidate them when relevant data changes.	Implemented
FR-401	Ingestion	Import jobs from a supported provider or controlled JSON payload.	Implemented
FR-402	Ingestion	Normalize provider data to a common job schema.	Implemented
FR-403	Ingestion	Prevent duplicate listings using provider ID, URL or title-company-location fallback.	Implemented
FR-404	Ingestion	Track provider availability, import failure and job freshness.	Partial / pilot gap
FR-501	Action	Save a job for the active user.	Implemented
FR-502	Action	Create or update application status using saved, applied, interview or rejected.	Implemented
FR-503	Action	Display tracked jobs ordered by recent activity.	Implemented
FR-601	Analytics	Capture search, result impression, explanation open, save, apply, status update and feedback events.	Pilot gap

Non-functional requirements

NFR	Requirement	Pilot target
NFR-01 Performance	Cached search responses should feel immediate; uncached search should complete within a usable interactive threshold.	p95 under 2 seconds cached; under 5 seconds uncached
NFR-02 Reliability	Provider or LLM failure must not crash the product or silently produce misleading results.	Actionable errors; deterministic fallback where supported
NFR-03 Security	Resume and profile data require authenticated access, secure transport and least-privilege storage.	No unauthenticated personal-data access
NFR-04 Privacy	Users must be able to understand, export and delete stored profile data.	Documented retention and deletion flow
NFR-05 Explainability	Every recommendation must expose its score components and source listing.	100% of ranked results

NFR	Requirement	Pilot target
NFR-06 Observability	Search, provider, cache, AI and application errors must be measurable.	Structured logs and operational dashboard
NFR-07 Accessibility	Core search, job review and tracking workflows must support keyboard and screen-reader use.	WCAG-oriented audit before beta

8. Product Flow & Release Criteria

The complete MVP loop and the conditions required to advance into a controlled pilot

Primary workflow

1. Profile	2. Intent	3. Interpret	4. Rank	5. Explain	6. Act	7. Track
Skills, experience, role and location	Natural-language search	Visible structured filters	Hybrid relevance score	Evidence and breakdown	Save, source or apply	Status progression

Planning assumptions and dependencies

Type	Dependency or constraint	Product response
AI provider	Query parsing and generated explanations may be unavailable or rate-limited.	Maintain deterministic parsing and explanation fallbacks.
Job providers	APIs may require credentials, change formats or become unavailable.	Isolate adapters and expose provider health.
Data quality	Job descriptions may omit salary, seniority or clear skills.	Preserve unknown values and avoid false certainty.
Privacy	Resume text is personal and potentially sensitive.	Require auth, consent, retention and deletion controls before pilot.
Small team	The MVP cannot solve every career workflow.	Protect the intent-to-action loop and defer adjacent automation.

PRD release criteria

- All Must requirements pass functional tests.
- Authentication and profile deletion are complete before external pilot access.
- A versioned relevance benchmark exists for representative search and profile cases.
- No critical privacy or cross-user data-isolation issue remains.
- Provider failure, empty results and AI failure have usable error states.
- Analytics events are validated end to end.
- A controlled rollback and incident owner are defined.

Plan phase decision

GO TO DEVELOP

The MVP scope is coherent and the highest-value loop is narrow enough to build. Development should proceed with explicit traceability to the requirements above and with pilot gates for security, analytics, freshness and AI quality.

PHASE 2 - PLAN

AI Strategy & Evaluation Plan

AI responsibility boundaries, hybrid ranking, fallbacks, evaluation and responsible deployment

PHASE
DECISION

GO

9. AI Product Strategy

Use AI where interpretation creates value; keep ranking evidence and user action controllable

AI responsibility boundary

Task	Method	Why
Interpret natural-language query	LLM structured JSON with heuristic fallback	Language is variable and benefits from contextual understanding.
Represent job and profile meaning	Embeddings	Captures semantic relationships beyond exact keywords.
Retrieve semantic candidates	Vector-search adapter	Separates retrieval from ranking and allows future index replacement.
Calculate fit score	Deterministic weighted formula	Supports repeatability, evaluation and transparent tuning.
Generate concise explanation	LLM grounded in supplied signals with deterministic fallback	Turns score components into readable reasoning without owning the score.
Choose whether to apply	Human decision	Career actions remain consequential and user-controlled.
Track application state	Deterministic workflow and database constraints	Status management does not require probabilistic AI.

AI processing workflow

Input	Understand	Retrieve	Score	Explain	User decision
Query or profile	Role, skills, level, location, remote	Keyword + semantic candidates	Weighted fit signals	Grounded reason and breakdown	Open, save, apply or ignore

Why hybrid rather than LLM-only

- **Repeatability** - The same signals can be compared across versions and evaluated against labels.
- **Control** - Product can tune weights without depending on prompt wording for every ranking decision.
- **Cost** - The generative model is not required to score every pair independently.
- **Transparency** - Users and the product team can inspect semantic, skill, keyword and location contributions.
- **Resilience** - Useful behavior remains available when the LLM is unavailable.

10. Ranking & Explainability

Implemented scoring logic, interpretation rules and known limitations

Implemented ranking formula

MATCH SCORE

Score = $0.50 \times \text{semantic similarity} + 0.20 \times \text{skill match} + 0.20 \times \text{keyword match} + 0.10 \times \text{location match}$. A seniority match can add a small adjustment to the keyword component. The final score is capped and shown as a percentage.

Signal	Weight	Meaning	Known limitation
Semantic similarity	50%	Embedding closeness between query/profile context and normalized job text.	May overvalue generally related language.
Skill match	20%	Overlap between extracted desired skills and job tokens.	Token overlap may miss synonyms or depth of experience.
Keyword match	20%	Overlap among role, skill and seniority terms.	Job descriptions can repeat keywords without true role fit.
Location match	10%	Location substring or compatible remote status.	Eligibility and time-zone constraints need richer modeling.
Seniority adjustment	Small	Strengthens an exact seniority match.	Seniority labels are inconsistent across providers.

Explainability contract

Output	Requirement
Match percentage	Derived from the deterministic formula; never invented by the language model.
Score breakdown	Expose semantic, skill, keyword and location components.
Matched skills	Show only skills supported by the extraction and job text.
Natural-language reason	Use only job context, profile context and calculated signals; keep it concise.
Source	Preserve the original provider and listing URL.
Uncertainty	Do not imply eligibility, guaranteed fit, hiring likelihood or salary certainty.

Fallback behavior

Failure	Fallback	User experience
No OpenAI key or parsing failure	Heuristic extraction of common skills, levels, locations and role terms.	Search remains available with reduced language flexibility.
Explanation generation failure	Template based on matched skills, location, seniority or semantic similarity.	The result still includes a grounded reason.
No filtered jobs	Fall back to broader job set before ranking.	Avoid an empty experience while preserving ranking evidence.
Redis unavailable	Continue without cache when safely implemented.	Higher latency but no loss of core function.
Provider failure	Return provider-specific actionable error and keep existing jobs usable.	One source does not break the whole product.

11. AI Evaluation Framework

Quality metrics, benchmark design, release gates and continuous improvement

Evaluation layers

Layer	Question	Metric
Query interpretation	Did the system extract the user's intended role, skills, level, location and remote preference?	Field-level precision, recall and exact/partial match
Retrieval	Did the candidate set contain the relevant jobs?	Recall@K and provider coverage
Ranking	Were the strongest jobs near the top?	NDCG@K, MRR, Precision@K and pairwise preference
Explanation	Was the reason supported, useful and concise?	Groundedness, unsupported-claim rate and usefulness rating
Product trust	Did users understand and act on results appropriately?	Explanation open rate, relevance feedback and save/apply conversion
Operations	Can the system deliver quality within cost and latency constraints?	p95 latency, failure rate and cost per ranked session

Benchmark design

- Create representative query sets across roles, seniority levels, locations, remote intent and transferable skills.
- Create candidate profiles with distinct experience and preference combinations.
- Label relevant and non-relevant jobs using a written rubric rather than intuition alone.
- Store model, prompt, embedding version, weights and provider snapshot with every benchmark run.
- Review disagreement cases and separate retrieval failures from ranking failures.
- Add adversarial and low-information cases, including vague queries and incomplete job descriptions.

Proposed pilot thresholds

Metric	Target	Decision use
Filter extraction field accuracy	≥ 90% on benchmark	Blocks launch if role or location interpretation is unreliable.
NDCG@10	Baseline plus ≥ 10% versus keyword-only	Tests whether hybrid ranking adds measurable relevance.
Top-5 irrelevance rate	≤ 15%	Protects candidate attention.
Unsupported explanation claims	< 2%	Trust and safety gate.
Search technical success	≥ 99% excluding provider outages	Operational reliability gate.
p95 uncached search latency	< 5 seconds	Usability guardrail.
Cost per search session	Set after baseline	Viability and model-routing input.

Responsible AI and privacy controls

Risk	Control
Sensitive resume data	Explicit consent, encryption in transit and at rest, access isolation, export and deletion.
Bias or proxy discrimination	Do not use protected attributes as ranking signals; audit outcomes and feedback by relevant cohorts where lawful.
Overconfident career advice	Use recommendation language, expose source data and avoid guarantees.
Hallucinated explanations	Ground the explanation in supplied context and calculated signals; benchmark unsupported claims.
Stale listings	Track import and update time; expire or flag old jobs.
Automation harm	Require user review before any future application or outbound communication.

Risk	Control
Prompt injection in job text	Treat listing content as untrusted data; separate instructions from content and test adversarial cases.

AI strategy decision

GO WITH GUARDRAILS

The hybrid architecture is appropriate for an MVP because it combines flexible language understanding with transparent, testable scoring. Pilot launch remains blocked until a versioned benchmark, privacy controls and analytics prove the system can be trusted and improved.

PHASE 3 - DEVELOP

As-Built Product & Architecture Report

What exists in the repository, how it maps to the PRD and what remains before a safe pilot

PHASE
DECISION
ITERATE

12. As-Built Product

Repository-backed implementation rather than proposed future capability

Implemented user-facing capabilities

Experience	As-built behavior
Homepage	Communicates natural-language search, ranking, explanation and application organization.
Search	Accepts natural-language queries, displays provider counts, allows job import and shows extracted filters.
Job results	Shows match percentage, location, seniority, remote status, tags, provider, explanation and score breakdown.
Profile	Captures name, location, skills, preferred roles, experience and pasted resume text.
Recommendations	Ranks the current job inventory against the stored profile.
Actions	Allows save, apply and original-source navigation.
Application tracker	Supports saved, applied, interview and rejected status updates.

Implemented backend capabilities

Area	As-built behavior
API	FastAPI endpoints for health, stats, import, search, job detail, profile, recommendations, save/apply and saved jobs.
Persistence	SQLAlchemy models for jobs, profiles, applications and saved-job relationships.
Search	Structured filters, optional PostgreSQL full-text search, embeddings and vector-search adapter.
Ranking	Deterministic hybrid score with explanation generation.
Ingestion	Adapters for JSON, Remotive, Adzuna, Greenhouse and Lever.
Caching	Redis-backed search and recommendation cache with targeted invalidation.
Delivery	Docker Compose for PostgreSQL/Redis and GitHub Actions for backend smoke and frontend build.

13. Technical Architecture

System boundary, core components, data flow and deployment model

Architecture view

Candidate	Next.js web app	FastAPI service	Data & AI services	External systems
Search, profile, review and tracking	Pages, components and API client	Business logic and REST endpoints	PostgreSQL, Redis, embeddings, ranking and fallbacks	OpenAI and job-provider APIs

Primary data flow

1. A provider adapter retrieves job data and normalizes each listing into the shared Job schema.
2. The backend deduplicates the listing and generates an embedding from normalized job text.
3. A candidate query is parsed into structured filters using the LLM or heuristic fallback.
4. The system retrieves candidate jobs using database filters and optional PostgreSQL full-text search.
5. The query or profile is embedded and compared with job vectors through the vector-search adapter.
6. The ranking service calculates the hybrid score and requests a grounded explanation.
7. The frontend presents the result and records candidate actions in the application tables.
8. Redis caches repeated search and recommendation responses and is invalidated when underlying data changes.

Core data model

Entity	Important fields	Product purpose
Job	Title, company, location, salary, description, requirements, tags, seniority, remote, URL, provider ID, source, embedding	Normalized opportunity and ranking input.
UserProfile	Name, skills, experience, preferred roles, location preference, resume text, embedding	Persistent personalization context.
Application	User, job, status, notes, optional match score, timestamps	Candidate pipeline state.
SavedJob	User, job, creation time	Explicit saved relationship and deduplication.

Key architecture decisions

Decision	Reason	Trade-off
FastAPI + Next.js separation	Clear service boundary and independently deployable frontend/backend.	More deployment and CORS configuration than a monolith.
PostgreSQL source of truth	Relational integrity for users, jobs and application state.	Embedding storage as text is an MVP compromise.
Redis cache	Reduces repeated ranking and AI work.	Requires invalidation discipline and failure handling.
Provider abstraction	Normalizes inconsistent external job data.	Provider-specific quality still varies.
Vector adapter	Allows pure-Python fallback and future FAISS/Pinecone replacement.	Current persistence and scale are limited.
Local fallback logic	Maintains useful behavior without external AI.	Less flexible interpretation than a capable model.

14. Requirements Traceability

How the implemented product maps to the essential PRD

Product capability	Requirement	Implementation evidence	Assessment
Profile context	FR-101, FR-102	Profile page and /user/profile endpoint	Complete for MVP
Natural-language intent	FR-201, FR-202	Search box and AIService.parse_query	Complete for MVP
Visible interpretation	FR-203	Search page extracted-filter panel	Complete
Hybrid ranking	FR-204	matching.py rank_jobs	Complete; benchmark pending
Explanation	FR-205	AIService.explain_match and JobCard breakdown	Complete; quality evaluation pending

Product capability	Requirement	Implementation evidence	Assessment
Recommendations	FR-301, FR-302	/recommendations endpoint and Redis cache	Complete
Provider ingestion	FR-401 to FR-403	JobIngestionService and upsert logic	Complete with provider constraints
Save/apply/status	FR-501 to FR-503	Application endpoints and tracker page	Complete
Account security	FR-103 / NFR-03	No production authentication shown	P0 gap
Analytics	FR-601 / NFR-06	No product analytics event layer shown	P0 gap
Freshness	FR-404	Timestamps exist; expiry workflow not shown	P1 gap

Delivery and CI evidence

- The repository includes Docker Compose services for PostgreSQL 16 and Redis 7 with persistence and health checks.
- GitHub Actions runs on pushes and pull requests to the default branch.
- The backend job installs dependencies, compiles the application and runs an API smoke test.
- The frontend job installs locked dependencies and executes a production build.
- Vercel and Railway deployment checks were passing when the repository was reviewed.

15. Release Readiness Assessment

What can be demonstrated now and what blocks an external pilot

Readiness scorecard

Dimension	Current evidence	Assessment
Core workflow	Search, explain, profile, recommend, save/apply and track are implemented.	Ready for demo
Technical deployment	Frontend and backend deployment checks pass; CI is defined.	Ready for controlled environment
AI quality	Logic is explicit, but no versioned benchmark result is included.	Not release-ready
Security and privacy	CORS and environment config exist; production authentication and user data controls are not demonstrated.	Blocked
Analytics	No end-to-end product event taxonomy or dashboard is implemented.	Blocked
Job freshness	Import/update timestamps exist, but automatic expiry and quality monitoring are not demonstrated.	Iterate
UX resilience	Some failures use alerts or generic messages; structured error recovery is limited.	Iterate
Accessibility	Basic labels exist; no completed accessibility audit is shown.	Iterate

P0 pilot backlog

Priority	Item	Evidence required
P0	Implement authentication and per-user authorization.	Cross-user isolation tests and security review.
P0	Add profile update, export and deletion.	Functional tests and privacy notice.
P0	Instrument the core event funnel.	Verified analytics payloads and dashboard.
P0	Create relevance and explanation benchmark runner.	Versioned baseline report.
P0	Add provider health, freshness and expiry handling.	Operational dashboard and stale-job test.
P0	Improve failure and empty-state UX.	Usability test across provider, AI and network failures.
P1	Persist ranking version and score provenance.	Reproducible recommendation record.

Priority	Item	Evidence required
P1	Complete accessibility audit and fixes.	Audit report and keyboard/screen-reader checks.
P1	Add structured notes and reminder fields.	Application workflow usability evidence.

Develop phase decision

ITERATE BEFORE PILOT

The repository is a credible, functional portfolio MVP and can be demonstrated publicly with non-sensitive test data. It should not be treated as production-ready for real resume data until P0 security, privacy, analytics, freshness and evaluation gaps are resolved.

DEMO READY CORE LOOP Search, explain, recommend and track	PILOT BLOCKED P0 CONTROLS Authentication, privacy and analytics	NEXT GATE VALIDATION Benchmark, pilot and evidence review
--	--	--

PHASE HANDOFF Development has established technical feasibility. The Launch phase is therefore framed as controlled evidence collection, not as a claim of production readiness.

PHASE 4 - LAUNCH

Pilot Measurement & Roadmap Plan

Controlled validation, analytics, experiments, release gates and evidence-based roadmap

PHASE
DECISION

CONDITIONAL

16. Pilot Objective & Cohort

Validate candidate value and trust before investing in advanced automation or commercial scale

Pilot objective

Determine whether the product helps active technology job seekers identify more actionable opportunities with less manual review and sufficient trust to return and manage their application workflow.

Recommended pilot design

Element	Recommendation
Cohort	20-30 active job seekers across product, engineering, data and AI roles; include career switchers and remote/regional candidates.
Duration	Four weeks, with weekly product reviews and a final decision meeting.
Data	Real candidate profiles only after authentication, consent and deletion controls are complete.
Job supply	A monitored mix of supported providers with documented freshness and availability.
Research	Onboarding observation, weekly interviews, relevance feedback and end-of-pilot survey.
Comparison	Keyword-only baseline or existing job-board workflow for selected tasks.
Decision	Scale, iterate or stop based on value, trust, repeated use, operational cost and risk.

Pilot hypotheses

ID	Hypothesis	Primary evidence
H1	Natural-language search reduces search setup effort for complex intent.	Task time, reformulation rate and usability feedback
H2	Hybrid ranking produces more relevant top results than keyword-only search.	NDCG@10, pairwise preference and top-5 irrelevance
H3	Explanations increase confidence without creating excessive review time.	Trust rating, explanation use and save conversion
H4	Profile recommendations create value beyond one-off search.	Recommendation-to-save rate and weekly return
H5	Integrated tracking increases repeat product use.	Status updates, active weeks and retained users

17. Measurement System

North Star Metric, funnel, quality guardrails and operational health

North Star Metric

HIGH-INTENT OPPORTUNITIES ACTED ON PER ACTIVE USER PER WEEK

A high-intent opportunity counts when a user saves a job, opens the source after reviewing the result, or marks the job as applied. The metric measures whether the product converts ranking into candidate action rather than generating searches alone.

Metric hierarchy

Acquisition	Activation	Relevance	Trust	Action	Retention	Operations
Eligible users and profiles	First successful search and viewed result	Top-K quality and feedback	Explanation use and confidence	Save, source open and apply	Active weeks and pipeline updates	Latency, failures, cost and freshness

Core metric dictionary

Metric	Definition	Pilot target
Activation rate	User creates a profile, completes a search and opens at least one result.	≥ 70%
Search-to-save	Search sessions producing at least one saved job.	≥ 15%
Search-to-apply	Search sessions producing an applied status.	Baseline; target after observation
Top-5 irrelevance	Share of top-five results marked irrelevant.	≤ 15%
Explanation inspection	Share of result viewers opening the score breakdown.	≥ 25%
Recommendation-to-save	Profile recommendations producing a saved job.	≥ query-only baseline
Week-4 retained use	Activated users performing a core action in week four.	≥ 35% pilot hypothesis
Provider freshness	Visible listings inside the accepted freshness window.	≥ 95%
Technical success	Core API requests completing without product error.	≥ 99%
Unsupported explanation rate	Explanations containing claims not supported by context/signals.	< 2%

Guardrails

- No sensitive profile data is visible across user accounts.
- Users can delete profile and resume content without support intervention.
- No automated application or communication occurs without explicit user review.
- Provider outages do not produce misleading freshness or match claims.
- Improved conversion does not come from lower relevance or manipulative urgency.
- AI quality, latency and cost are reviewed together rather than optimized in isolation.

18. Analytics & Experiment Plan

Events, dashboards and tests required to turn product usage into decisions

Essential event taxonomy

Event	Trigger	Key properties
profile_created	Profile saved successfully	Profile completeness, source, consent version
search_submitted	Natural-language search sent	Query length, parsed fields count, search version
filters_extracted	Structured intent returned	Role/skills/location/level/remote and parser type
results_viewed	Ranked list rendered	Result count, provider mix, latency and ranking version

Event	Trigger	Key properties
job_opened	Job detail opened	Rank, score band, source
explanation_opened	Score breakdown expanded	Rank and score components
job_saved	Save completed	Rank, score band and source
source_opened	Original listing opened	Rank, source and freshness age
application_status_updated	Status changed	Old status, new status and time since save
relevance_feedback	User rates result	Relevant/irrelevant and reason
provider_import_completed	Import finishes	Provider, imported, updated, failed and duration
product_error	User-visible error occurs	Flow, provider/model, status and recoverability

Priority experiments

Experiment	Hypothesis	Primary metric	Decision
Hybrid vs keyword ranking	Hybrid ranking improves actionable relevance.	Search-to-save and NDCG@10	Retain, retune or simplify ranking.
Explanation vs score only	Grounded explanations improve trust and action.	Save conversion and trust rating	Keep, redesign or reduce explanation.
Profile vs query only	Persistent context improves recommendations.	Recommendation-to-save	Invest in profile depth or reduce setup.
Resume text optionality	Resume context improves transferable-role discovery.	Top-5 relevance by persona	Keep optional, redesign or exclude.
Result density	A smaller high-confidence list reduces overload.	Decision time and save rate	Choose default result count.

Review cadence

Cadence	Review
Daily during pilot	Errors, provider health, freshness, AI failure, latency and privacy incidents.
Weekly	Funnel, relevance feedback, user interviews, benchmark drift and experiment status.
End of pilot	Value, trust, retention, operations, cost, risk and product-market evidence.
Before every release	Requirements traceability, AI benchmark, security, privacy, accessibility and rollback readiness.

19. Launch Gates & Product Roadmap

Evidence required for controlled release and sequence of future investment

Controlled pilot gate

Gate	Required evidence
Security and privacy	Authentication, authorization, deletion, consent and no critical data-isolation defect.
AI quality	Versioned benchmark meets extraction, ranking and explanation thresholds.
Product quality	Core loop passes end-to-end tests with actionable errors and empty states.
Measurement	Event taxonomy and dashboard are validated.
Job supply	Provider health, freshness and duplicate monitoring are operational.
Operations	On-call owner, rollback plan, incident severity and support channel are defined.

Evidence-gated roadmap

Horizon	Outcome	Capabilities	Evidence gate
Now - Pilot readiness	Safe measurable validation	Auth, analytics, benchmark, feedback, privacy, freshness and resilience	All P0 gates pass
Next - Workflow depth	Improve repeated use	Notes, reminders, timeline, alerts and richer profile management	Week-4 retention and tracking use
Later - Career assistance	Improve application quality	Resume tailoring, gap analysis and interview preparation	Demonstrated trust and willingness to pay
Future - Controlled agent	Reduce repetitive application work	Human-reviewed form assistance and communications	Legal, safety and quality validation
Not committed	Employer-side platform	Recruiter workflows, ATS and enterprise administration	Separate market evidence and strategy

Scale / iterate / stop framework

Decision	When to choose it
Scale	Strong relevance improvement, trusted explanations, meaningful action conversion, retained workflow use and manageable operating cost.
Iterate	Users value the problem and parts of the workflow, but ranking, setup, provider quality or trust prevents repeat use.
Stop or reframe	The product does not outperform simpler search, users do not return, or sustainable job supply and privacy economics are not viable.

Launch phase recommendation

CONDITIONAL PILOT

Do not add more AI features before validating the current loop. Complete the P0 gates, run a small controlled pilot, and make the next roadmap decision from relevance, action, retention, trust, operational cost and privacy evidence.

Product Manager Reflection

What changed, the most important trade-offs and how the project demonstrates AI product leadership

How the product thesis evolved

The initial idea could have become a generic AI job-search interface. The stronger product thesis emerged by focusing on candidate attention and decision confidence rather than job inventory. That shift changed the product from a search box with AI into a connected workflow: interpret intent, rank transparently, explain the evidence, support action and learn from outcomes.

Most important product decisions

Decision	Why it matters
Use hybrid scoring instead of LLM-only ranking	Preserves transparency, repeatability, evaluation and cost control.
Separate ranking from explanation	Prevents fluent text from becoming the hidden source of the numeric decision.
Make extracted filters visible	Allows the user to inspect how natural language was interpreted.
Include tracking in the MVP	Creates workflow continuity and measurable action rather than an isolated recommendation demo.
Defer automatic applications	Protects candidate control and avoids high-risk automation before trust is established.
Label outcomes honestly	A functional repository proves feasibility, not product-market fit or business success.

What I would do differently next

- Begin primary candidate interviews and benchmark-labeling before expanding provider coverage.
- Implement authentication and analytics earlier because both are prerequisites for trustworthy product learning.
- Persist ranking version and provenance from the first implementation to simplify evaluation and debugging.
- Model location eligibility, compensation and seniority more explicitly rather than relying on text normalization alone.
- Design feedback collection as part of the result card, not as a later analytics enhancement.
- Use a provider-quality score so supply decisions are driven by freshness, completeness and relevance rather than raw listing count.

Portfolio evidence of AI product management

Competency	Evidence in this case study
Problem framing	Reframed from job inventory to candidate attention and decision support.
Product strategy	Defined beachhead users, value proposition, product principles, assumptions and phased roadmap.
Requirements	Converted the concept into a bounded MVP, user stories, requirements, NFRs and release gates.
AI design	Defined model responsibilities, deterministic boundaries, scoring, fallbacks, evaluation and responsible AI.
Technical fluency	Connected frontend, API, database, cache, embeddings, providers and CI/CD decisions to product outcomes.
Measurement	Defined North Star, funnel, event taxonomy, benchmark metrics, experiments and scale decisions.
Integrity	Separated actual implementation from targets, recommendations and unvalidated outcomes.

Final recommendation

PORTFOLIO COMPLETE - PRODUCT VALIDATION NEXT

Use the current MVP as evidence of product thinking, AI solution design and delivery capability. The next investment should be a safe, measurable candidate pilot—not a larger feature set. The product earns the right to scale only after it demonstrates relevance improvement, trusted action and repeat workflow value.

Appendix - Repository Evidence Map

Implementation references used to ground the portfolio document

Evidence area	Repository reference	What it demonstrates
Product overview and setup	README.md	Stack, job sources, AI mode, local setup, infrastructure and CI/CD summary.
API and orchestration	backend/app/main.py	Search, profile, recommendations, ingestion, save/apply and status endpoints.
Hybrid ranking	backend/app/services/matching.py	Score weights, matched skills, breakdown and explanation call.
AI parser and explanation	backend/app/services/ai.py	LLM structured extraction, heuristic fallback and grounded explanation.
Provider normalization	backend/app/services/ingestion.py	Remotive, Adzuna, Greenhouse and Lever adapters.
Persistence model	backend/app/models.py	Jobs, profiles, applications, saved jobs and uniqueness constraints.
Homepage and value proposition	frontend/app/page.tsx	Natural-language search and product workflow message.
Search results	frontend/app/search/page.tsx	Provider counts, imports, extracted filters and ranked results.
Result explanation and actions	frontend/components/JobCard.tsx	Score, breakdown, save, apply and original source.
Profile experience	frontend/app/profile/page.tsx	Skills, experience, role, location and resume context.
Recommendations	frontend/app/recommendations/page.tsx	Profile-ranked job feed.
Application tracker	frontend/app/saved/page.tsx	Saved, applied, interview and rejected state management.
Delivery pipeline	.github/workflows/ci.yml	Backend compile/smoke checks and frontend production build.
Local infrastructure	docker-compose.yml	PostgreSQL and Redis services with health checks.

Public portfolio links

Repository: [AI Job Search Agent on GitHub](#)

Interview summary

I designed and built an explainable AI job-search MVP that converts natural-language career intent into structured filters, combines semantic and deterministic ranking, generates grounded match explanations, personalizes results from candidate profiles and supports application tracking. I treated the repository as technical feasibility evidence, then defined the missing evaluation, security, analytics and launch gates required before claiming real product outcomes.

Confidentiality and evidence note

This is a public portfolio case study based on a public repository. Product targets, pilot thresholds, personas, business hypotheses and future roadmap items are planning artifacts. They must not be interpreted as observed customer results, production performance, revenue or market validation.

AI JOB SEARCH AGENT

Product thinking made visible through evidence

Min Thu Kyaw • AI Product Manager • July 2026