CareerOS — Live Pipeline

RUNNING ON NVIDIA NIM (FREE) · LLAMA-3.1-70B · [redacted] COMPLIANCE · [redacted]

ENTRY — DAILY WORKFLOW

the complete job search pipeline — scout → add → process → review → outreach → apply

`python3 main.py --phase scout` `python3 job_adder.py` `./process_and_review.sh` `python3 agent_outreach.py` `python3 market_align.py`

careeros2/ · NVIDIA NIM API · meta/llama-3.1-70b-instruct

CONFIGURATION — CONFIG.PY

CandidateProfile

[redacted]

location: New York, NY
skill_anchors: 50+ terms
cv_summary: research engineer framing
target_domains: 15 sectors

config.py

[redacted]_KNOWN

80+ named institutions
universities · teaching hospitals
nonprofit research orgs
government labs · NIH · NSF
Agent 3B imports directly

config.py

TARGET_CAREERS_PAGES

70+ URLs across:
STEM boards · gov labs · nonprofits
public health · UX/HF boards
scicomm · academic · ATS boards
Brian's Job Search (manual)

config.py

PHASE: SCOUT — PYTHON3 MAIN.PY --PHASE SCOUT

1 Agent 1 — Title Dictionary

agents/agent1_title_dict.py

Reads config skill_anchors + target_domains.
Builds: primary, secondary, exploratory titles.
30 role variants. Saves title_dictionary.json.

2 Agents 2A–2E — Scout Queries

agents/agents2_scout.py

2A: ATS boards (Greenhouse, Lever, Ashby)
2B: Aggregators (LinkedIn, Indeed, Google Jobs)
2C: X-ray + pre-posting signals
2D: Startups (Wellfound, Seed-Series B)
2E: 70+ target careers pages

outputs search queries → output/scout_queries.json → human runs in browser

HUMAN STEP — FIND + ADD JOBS

▮ Manual Scouting

Brian's Job Search (briansjobsearch.com)
LinkedIn notifications + alerts
CUNY CareerPlan STEM boards
Handshake (CUNY credentials)
Direct careers pages from config

job_adder.py running

job_adder.py · localhost:7777

Browser form at localhost:7777.
Paste full messy JD text — auto-cleaned.
Appends to output/postings.json.
One click per job. No JSON editing.

output/postings.json
`[ { title, company, location, url, source, jd_text, remote } ]`

PHASE: PROCESS — ./PROCESS_AND_REVIEW.SH

Pre-dedup + dual filter

Agent 4 quick dedup pass first.
Dual filter: title match OR ≥2 skill anchors.
Freshest postings processed first.

Sequential per posting: 3A → 3B → 3C

agents/agents3_filter_intel.py

Each posting runs all three agents in order.
3C skipped if Tier C or BLOCKED.
Results saved to postings_processed.json.

sequential — per posting

3A Fit Scorer

Score 1-10, cites exact JD language.
PhD required: -2.0, HARD_BLOCK.
PhD preferred: -0.5 only.
Multiple locations extracted.
Commensurable experience mapped.
Sets Tier A(≥8) / B(≥6) / C.

3B [redacted] Compliance

[redacted] **by rule:** university, hospital, nonprofit, gov lab.
[redacted] **by name:** 80+ known institutions from config.
CLEAR / POSSIBLE / RISKY / BLOCKED.
E-Verify [redacted]

3C Company Intel

Runs only if Tier A/B + not BLOCKED.
Builds: why_this_hire, team structure, hiring speed estimate.
3 validation project options.
Application angle for candidate.

Agent 4 full dedup pass → saves output/postings_processed.json

Tier A — apply now
score ≥8, [redacted] not BLOCKED

Tier B — review
score 6-7.9, [redacted] not BLOCKED

Tier C — monitor
score <6

BLOCKED
explicit no [redacted]

REVIEW + EXPORT — ./PROCESS_AND_REVIEW.SH (CONTINUED)

review.py running

review.py

Human-readable terminal output.
Per job: tier, visa, salary, reports-to,
why this hire, quals you match/lack,
CV evidence, red flags, hiring speed,
validation projects, how to position.

export_excel.py running

export_excel.py

Color-coded Excel tracker.
Tier A green · Tier B amber · Tier C red.
Auto-adds new columns as new fields appear.
Preserves manually added columns.
Summary sheet with tier counts.

output/job_tracker.xlsx
output/postings_processed.json
mark liked jobs → Status = "liked"

MARKET ALIGNMENT — PYTHON3 MARKET_ALIGN.PY

market_align.py running

market_align.py

Reads liked / Tier A jobs from processed JSON.
Extracts domain keywords + high-value technical terms.
Suggests new skill_anchors for config.py.
Suggests role titles to add to scout queries.
Prints ready-to-paste config.py block.
CV framing suggestions by domain angle.

▮ Update config.py

Paste suggested skill_anchors into config.py.
Re-run scout to regenerate queries
with updated profile framing.
Repeat weekly as new jobs are processed.

OUTREACH — PYTHON3 AGENT_OUTREACH.PY

agent_outreach.py — contact identification + message generation running

agent_outreach.py · output/outreach_targets.json

Who to find
[redacted]r / team [redacted]
Name hint from JD text
Target department

How to find (ordered)
1. theorg.com org chart
2. Direct website team pages*
3. Google name + title
4. LinkedIn X-ray (verify!)
5. hunter.io email finder
*when direct URL available

What to send
LinkedIn note (<280 chars)
Cold email (150-200 words)
Follow-up angle (7 days)
VVP suggestion per role
Leads with their work, not your need

▮ HUMAN: verify before sending

Confirm name is real and currently affiliated · theorg.com + company website are ground truth · LinkedIn X-ray = starting point only, not verified list
Short company names prone to false X-ray matches · Send LinkedIn note OR cold email — not both at once · 5 contacts per day (Cultivated Culture cadence)

ACTION — APPLICATION MATERIALS

5 Cover Letter

Written manually for Tier A roles.
Addressed to named PI when known.
Para 1: technical match + tools.
Para 2: diagnostic thinking example.
Para 3: lab-specific connection.
Para 4: visa, availability, relocation.
No personal statement in CV.

6 CV Selection

1-page CV: ATS portal submission.
2-page CV: direct email to [redacted]
Both: zero columns, zero italics,
zero tables. Pure ATS-safe text.
Skills first · Experience · Education last
(for non-academic roles).

7 VVP

Value Validation Project.
Built on public data, shareable.
Sent to contact before or alongside
formal application.
Neuro-LLM project · HF project
Pipeline docs already available.

▮ Two-track submission

Track 1: Email directly to [redacted] — cover letter + 2-page CV.
Track 2: Apply through careers portal — 1-page CV.
Both tracks simultaneously for every Tier A role.
Subject line: [Role] Application — [One differentiating phrase].

output/[redacted]_CoverLetter.docx
output/Vaidehi_Patel_1page.pdf
output/Vaidehi_Patel_2page.pdf
output/[redacted]Interview_Prep.docx

CURRENT PIPELINE STATE — MARCH 2026

live stats postings processed: 14 · tier A: 10 · tier B: 4 · [redacted] blocked: 0 · CLEAR: 7 · POSSIBLE: 7 — top matches: [redacted] Lab (9.0) · [redacted] Research Manager (9.0) · [redacted] Research Assistant (8.5) · [redacted] Fellow (8.5, $91K) — applications in progress: [redacted] (cover letter done) · [redacted]orwarded to HR by [redacted])

SELF-REFINEMENT LOOP — WEEKLY

market_align.py

After marking liked jobs.
Update skill_anchors.
Re-scout with new framing.

config.py update

Add new skill_anchors.
Add new career pages.
Adjust tier thresholds
if scoring feels off.

outcome tracking

Mark applied in Excel.
Mark responses received.
After 20+ applications:
pattern analysis of
what Tier A actually converts.

COMPLETE FILE MAP

```
# scripts you run
main.py --phase scout → generates search queries
job_adder.py → browser form, adds postings
process_and_review.sh → process + review + export + open Excel
market_align.py → keyword analysis of liked jobs
agent_outreach.py → contact targets + messages
review.py → human-readable terminal review
export_excel.py → color-coded job_tracker.xlsx
```

```
# output files
output/scout_queries.json ← phase scout
output/postings.json ← job_adder.py (human-written)
output/postings_processed.json ← phase process
output/job_tracker.xlsx ← export_excel.py
output/outreach_targets.json ← agent_outreach.py
output/title_dictionary.json ← phase scout (agent 1)
output/postings_template.json ← blank template for reference
```

entry / config · seed / title · scout (2A-2E) · filter / intel (3A-3C) · action / application · outreach agent · refinement loop · live / running · human gate · file output

CareerOS · NVIDIA NIM (free) · meta/llama-3.1-70b-instruct · [redacted]compliance · human-in-the-loop · no fabrication guards