

COIE

Career Outreach Intelligence Engine

Product Requirements Document • v2.0

Author	Kshitij Dhar
Version	2.0 - March 2026 (v1.0 in Feb 2026)
Status	Shipped - Running in Local Environment
Stack	Python · Playwright · FastAPI · React · SQLite · Groq · Hunter.io
Platform	Local (Windows) - Task Scheduler + uvicorn
Changes in v2.0	Indeed scraper, Hybrid scoring, Dynamic config UI, Excel export, Skipped jobs, India-filtered recruiter discovery

1. Executive Summary

COIE is a personal job search automation system built to **transform passive job browsing into a structured, data-driven outbound recruiting pipeline**. It scrapes job listings twice daily from LinkedIn, Naukri, and Indeed, scores each listing against a candidate's resume using semantic AI with experience and seniority-aware penalties, surfaces high-match opportunities in a real-time dashboard, discovers recruiter contact details on demand, and drafts personalized outreach emails - all running locally with zero ongoing cost.

Problem: The modern job search is broken for candidates. Manually browsing job boards, copy-pasting roles into spreadsheets, and sending generic applications is time-consuming, untracked, and low-converting. High-quality roles are missed. Outreach is inconsistent. There is no visibility into the pipeline.

Solution: COIE automates the top-of-funnel entirely - scraping, scoring, and surfacing the best matches - while keeping the user in control of decisions that matter: which jobs to pursue and what to say. The result is a structured pipeline with measurable conversion metrics.

2. Objectives & Success Metrics

2.1 Primary Objectives

- Automate daily job discovery across LinkedIn, Naukri, and Indeed for desired roles and locations
- Score job listings against your resume using semantic similarity with experience and seniority penalties
- Surface recruiters' contact information on demand without a manual LinkedIn search
- Generate personalized outreach emails that reflect genuine resume-JD alignment
- Provide a live dashboard for pipeline tracking from discovery to interview
- Enable dynamic configuration of keywords, locations, and thresholds without code changes

2.2 Success Metrics

Metric	Target	Current
Jobs scraped per run	60–100	~80 (3 sources)
Unique jobs after dedup	> 80% retention	~85%
High-match rate (70%+)	10–20%	~15%
Email generation success	> 90%	100%
Hunter recruiter find rate	> 60%	In progress
Pipeline reply rate	> 10%	Tracking started

3. User Persona

COIE is designed for a single primary user - an experienced professional actively searching for roles in the Indian startup and enterprise market.

Attribute	Detail
Role	Product Manager / Associate PM - 2–5 years experience
Market	Bengaluru, Hyderabad, Remote (India)
Job boards	LinkedIn, Naukri, Indeed - checks manually 1–2x per week
Pain points	Missing new postings, no tracking system, generic applications, no time for cold outreach
Goals	Land interviews at high-growth startups and established tech companies
Tech comfort	Comfortable running scripts, using dashboards, and reading logs

4. System Architecture

COIE consists of four independently running components that communicate through a shared SQLite database:

Component	Responsibility
scheduler/runner.py	Orchestrates the full pipeline. Reads config dynamically from data/config.json on each run. Triggered by Windows Task Scheduler at 09:00 and 18:00 daily. Resets seen_hashes.json every 7 days automatically.
Scraper Layer	Playwright-based scrapers for LinkedIn, Naukri, and Indeed. URL normalization + SHA256 hash deduplication across all three sources.
Scoring Engine	Sentence transformer (all-MiniLM-L6-v2) cosine similarity as base score. Multiplied by experience penalty (based on extracted years requirement) and seniority penalty (based on title keywords).
FastAPI Backend	Reads from SQLite, exposes REST endpoints for jobs, stats, config, status updates, Excel export, and on-demand outreach.
React Dashboard	Served as static files via FastAPI. Pipeline view, Job Feed with filters and export, Outreach Queue, Settings panel.
Recruiter Discovery	Hunter.io domain search API - triggered manually per job. India-location filtered to exclude US/UK recruiters.
Email Generator	Groq API (Llama 3.1 8B) - generates personalized 4–6 line outreach email from resume + JD + recruiter context.
Notification System	SMTP via Gmail - HTML email digest after each run with top 10 matches and direct apply links.

5. Pipeline Flow

5.1 Automated Pipeline (twice daily)

The following steps execute automatically at 09:00 and 18:00 IST:

- **Load config** - Reads data/config.json freshly on each run - picks up any changes made in the Settings panel
- **Reset hashes (if due)** - Clears seen_hashes.json every 7 days for a fresh scrape cycle
- **Cleanup stale jobs** - Deletes New/Skipped jobs older than 7 days from the database
- **Scrape** - LinkedIn, Naukri, and Indeed scraped for configured keywords and locations
- **Deduplicate** - URL normalization + SHA256 hash prevents duplicates across sources and runs
- **Store** - New unique jobs inserted into SQLite (coie.db)
- **Score** - Resume and JD vectorized, cosine similarity computed, experience and seniority penalties applied
- **Notify** - HTML email with top 10 matches, scores, and direct apply links sent to the inbox

5.2 On-Demand Outreach Flow (user-triggered)

- **Review Outreach Queue** - Jobs at or above the outreach threshold (configurable, default 75%)
- **Perform Outreach** - User clicks the button on a specific job card
- **Recruiter Discovery** - Hunter.io queries the company domain, returns an India-based HR/TA contact with a verified email
- **Email Generation** - Groq generates a personalized 4–6 line email with a subject line
- **Review & Send** - Draft displayed in card. User clicks 'Open in Gmail' - To, Subject, Body pre-filled
- **Track** - User marks job as 'Outreach Sent' - pipeline tracks from here through Replied → Interview

6. Feature Specifications

6.1 Scraper Layer

- Playwright-based headless browser scraping for LinkedIn, Naukri, and Indeed
- LinkedIn: dual-page pattern, GeoID filtering locks results to India. The extended geold map covers all major Indian cities with an India-wide fallback
- Naukri: headless=False with anti-detection init script. jobAge=7 freshness filter. Stale signal filtering drops jobs older than 2 weeks
- Indeed: headless=False, click-to-panel pattern extracts full JD from right panel on the listing page (avoids Cloudflare on direct viewjob URLs). fromage=7 filter. Defaults posted date to 'Today' since all results are within 7 days
- URL normalization strips tracking parameters before hashing - prevents the same job with different UTM params from being treated as distinct
- seen_hashes.json auto-resets every 7 days to ensure fresh scrape cycles

6.2 Scoring Engine

- **Model:** sentence-transformers/all-MiniLM-L6-v2 - 384-dimension semantic vectors, runs fully offline

- **Base score:** cosine similarity normalized from [-1,1] to [0–100%]
- **Experience penalty:** extracts required years from JD using regex (supports ranges, word numbers, minimums, verb phrases). Penalizes up to 45% for 6+ year gap vs CANDIDATE_YEARS
- **Seniority penalty:** 20% reduction for Senior/Lead/Director/VP titles when CANDIDATE_LEVEL is 'mid' or 'junior'. Checks both job title and description
- **Final score** = base × experience_multiplier × seniority_multiplier
- Log output shows full breakdown: 74.3% (base:74.3% exp_req:3yr exp_mult:1.0 sen_mult:1.0)
- Jobs with < 50 character descriptions skipped and shown in Unscored accordion

```
CANDIDATE_YEARS = 3      # set in scorer/scorer.py
CANDIDATE_LEVEL = 'mid'  # 'junior' | 'mid' | 'senior'
```

6.3 Dynamic Configuration

- All user-facing settings are editable from the Settings panel in the dashboard - no code changes needed
- Settings saved to data/config.json, read fresh on each pipeline run
- Configurable: keywords (add/remove), locations (add/remove), match threshold (slider), outreach threshold (slider)
- Changes take effect on the next scheduled run or the immediate manual run
- Unknown config keys ignored - prevents injection. Missing config.json falls back to defaults

```
{ "keywords": ["Product Manager"], "locations": ["Bengaluru"],
  "match_threshold": 75, "outreach_threshold": 75 }
```

6.4 Dashboard

- **Pipeline Overview:** stat cards (scraped, unique, high match, outreach sent, replied, interview rate), conversion funnel, top matches by source
- **Job Feed:** scored jobs with filters for score threshold, source (LinkedIn/Naukri/Indeed), location, and recency (Today/Last 3 days/This week). Excel export with color-coded scores
- **Skipped Jobs accordion:** manually skipped jobs accessible at the bottom of the feed, can be un-skipped
- **Unscored Jobs accordion:** jobs with no description (authwall or JS-rendered) - respects source/location filters
- **Outreach Queue:** jobs at or above outreach threshold. Each card shows recruiter info, draft email, status controls, and Gmail link
- **Settings panel** live config editing with save confirmation
- **Status pipeline:** New → Outreach Sent → Replied → Interview → Rejected/Skipped

6.5 Recruiter Discovery

- Hunter.io Domain Search API - triggered per job on user demand, not automatically
- India location filtering: post-fetch filter removes US/UK/Canada-based contacts. Falls back to all results if no India-based found
- Filters for HR department, personal emails only (not careers@/hiring@ generic inboxes)
- Prefers recruiter/talent acquisition titles over generic HR roles
- Only fires for jobs at or above the outreach threshold - conserves free tier credits (25/month)

6.6 Email Generation & Notification

- Outreach: **Llama 3.1 8B** via **Groq** (free). Input: job title, company, JD (1500 chars), resume (2000 chars), recruiter name, and title. Output: subject + 4–6 line body
- Digest email: top 10 high-match jobs (filtered to threshold+, excluding Skipped/Rejected). HTML table with role, company, score, source, and a direct apply button
- Dashboard link in email points to localhost:8000

7. Technology Stack

Layer	Technology	Rationale
Scraping	Playwright (Python)	Full browser automation, handles JS-heavy pages, anti-detection capable
Scoring	sentence-transformers + rule-based penalties	Local inference, no API cost, semantic > keyword matching, seniority-aware
Backend API	FastAPI + uvicorn	Async, fast, auto-generates Swagger docs, lightweight
Database	SQLite	Zero config, file-based, sufficient for personal use, portable
Frontend	React + Vite	Fast dev experience, served as static files via FastAPI
Email Generation	Groq API (Llama 3.1 8B)	Free tier, fast inference, sufficient quality for short structured emails
Recruiter Search	Hunter.io	Verified work emails, company domain resolution, free tier
Notifications	Python smtplib + Gmail	No dependencies, HTML email support, app password auth
Scheduling	Windows Task Scheduler	Native OS scheduler, no additional process, reliable
Logging	Loguru	Structured logs, automatic rotation, 7-day retention

8. Key Design Decisions

Decision	Rationale
Config in JSON, not DB	Human-readable, easy to backup/restore, editable from the dashboard UI without a migration system
Read the config on each run	Changes take effect without restarting uvicorn or the scheduler - only the next pipeline run is needed

seen_hashes auto-reset (7d)	Jobs stay fresh without manual intervention. DB history (statuses, outreach) preserved independently
Experience penalty uses the lower bound of the ranges	'7-10 years' → 7 is the minimum requirement. Penalizes fairly without over-penalizing stretch roles
Seniority penalty is less aggressive than the experience penalty	Keyword matching is fuzzier than year extraction. 80% multiplier vs 55% at the extreme - flags without eliminating
Indeed click-to-panel pattern	Visiting /viewjob directly triggers Cloudflare. Clicking a card on the listing page stays in the same authenticated session
URL normalization before hashing	Same job posted with different tracking parameters (UTM, referral) deduplicates correctly across scrapes
Skipped jobs accordion	Manually skipped jobs are removed from the main feed but remain accessible - can be restored without re-scraping
On-demand outreach	Hunter credits are only spent on jobs the user has consciously reviewed. Prevents waste of 25/month free tier
Local deployment	Resumes, API keys, and email drafts are sensitive. Local keeps data private and eliminates hosting costs

9. Known Limitations & Roadmap

9.1 Current Limitations

- LinkedIn descriptions are gated behind login (authwall) - only short snippets available, reducing score accuracy for LinkedIn jobs
- Indeed descriptions require JS panel to load after clicking - occasional timeouts result in unscored jobs
- Hunter.io free tier limited to 25 searches/month - constrains outreach volume
- Experience extraction regex may miss creative phrasings ('half a decade', non-English JDs)
- Config changes require the next pipeline run - not instant
- Single user by design - no multi-user support, authentication, or cloud deployment

9.2 Roadmap

Feature	Description	Priority
Follow-up sequencing	Auto-draft follow-up after 5 days of no reply	High
Resume gap analysis	Identify skills in top JDs missing from the resume	Medium
LinkedIn authenticated scraping	Cookie-based session for full JDs	Medium
Company research module	Auto-fetch funding, size, product info to enrich outreach	Medium
Supabase migration	Cloud DB for remote dashboard access	Low

Railway deployment	Full cloud deployment for an always-on pipeline	Low
--------------------	---	-----

10. Implementation Notes

Project Structure

```

COIE/
├── api/main.py           ← FastAPI endpoints + config API + Excel export
├── scraper/
│   ├── linkedin_scraper.py ← extended geoId map, dual-page pattern
│   ├── naukri_scraper.py  ← headless=False, anti-detection
│   ├── indeed_scraper.py  ← click-to-panel JD extraction
│   └── deduplicator.py     ← URL normalization + SHA256 hash dedup
├── scorer/
│   ├── scorer.py          ← semantic + experience + seniority scoring
│   └── resume_parser.py
├── recruiter/hunter.py    ← India-filtered recruiter discovery
├── outreach/generator.py  ← Groq email generation
├── scheduler/
│   ├── runner.py          ← dynamic config + seen_hashes auto-reset
│   └── notify.py          ← HTML email digest (top 10 matches)
├── storage/
│   ├── database.py        ← auto-cleanup of stale jobs (7 days)
│   └── models.py
├── dashboard/src/App.jsx  ← React: Pipeline, Feed, Outreach, Settings
├── data/
│   ├── config.json        ← user settings (auto-created)
│   ├── coie.db            ← job database
│   └── seen_hashes.json    ← dedup memory (auto-resets every 7d)
├── config.py              ← static config (paths, schedule times only)
└── logs/                  ← Loguru rotating logs (7-day retention)

```

Configuration Reference

Variable	Default / Description
MATCH_THRESHOLD	75 - minimum score to appear in Job Feed and email digest (configurable via Settings)
OUTREACH_THRESHOLD	75 - minimum score for Outreach Queue and Hunter API calls (configurable via Settings)
KEYWORDS	Product Manager, Associate Product Manager (configurable via Settings)
LOCATIONS	Bengaluru, Hyderabad (configurable via Settings)
CANDIDATE_YEARS	Set in scorer/scorer.py - your total years of experience for penalties
CANDIDATE_LEVEL	Set in scorer/scorer.py - 'junior' 'mid' 'senior'

RUN_TIME1 / RUN_TIME2	09:00 / 18:00 - set in config.py
DB_PATH	data/coie.db
RESUME_PATH	scorer/resume.pdf

Windows Task Scheduler Setup

Pipeline Task

- Program: python
- Arguments: -m scheduler.runner
- Start in: C:\Users\[you]\Projects\COIE
- Trigger: Daily at 09:00 and 18:00
- Condition: uncheck 'Start only if on AC power'

API Server Task

- Program: cmd
- Arguments: /c "cd /d C:\Users\[you]\Projects\COIE && env\Scripts\uvicorn api.main:app --port 8000 --log-config api/log_config.json"
- Trigger: At log on, with 15 second delay
- Settings: check 'Run whether user is logged on or not', uncheck 'Start only if on AC power'

10. Screenshots

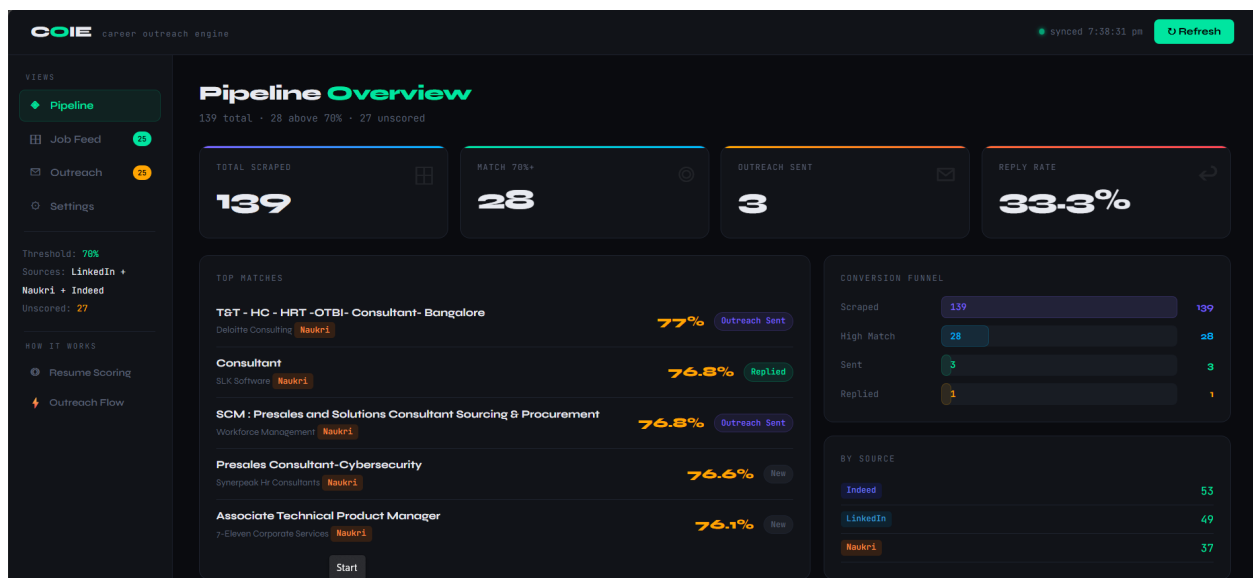


Fig: Pipeline Overview Page with Indeed jobs loaded

COIE

career outreach engine

synced 7:36:18 pm

Refresh

Views

Pipeline

Job Feed25

Outreach25

Settings

Threshold: 70%

Sources: LinkedIn + Naukri + Indeed

Unscored: 27

How it works

Resume Scoring

Outreach Flow

Job Feed

25 scored • 27 unscored

Filter: Above threshold (70%)

All sources

All locations

This week

Export

ROLE	COMPANY	SOURCE	MATCH SCORE	POSTED	STATUS
T&T - HC - HRT -OTBI- Consultant- Bang..	Deloitte Consulting	Naukri	77%	1 day ago	New
Consultant	SLK Software	Naukri	76.8%	1 day ago	New
SCM : Presales and Solutions Consultan...	Workforce Management	Naukri	76.8%	5 days ago	New
Presales Consultant-Cybersecurity	Synerpeak Hr Consultants	Naukri	76.6%	1 day ago	New
Associate Technical Product Manager	7-Eleven Corporate Services	Naukri	76.1%	5 days ago	New
Package Consultant-SAP HANA Sales	IBM	Naukri	75.2%	2 days ago	New
FULL STACK - JAVA Consultant	ProcureHR	Indeed	75%	Today	New
Package Consultant-SAP HANA Sales	IBM	Naukri	74.6%	2 days ago	New
Product Manager 3	Oracle	Naukri	74%	4 days ago	New
Associate - Product Manager	Anarock Property Consultants	Naukri	73.7%	1 day ago	New

Fig. The Job Feed Page with Export option

COIE

career outreach engine

synced 12:04:16 pm

Refresh

Views

Pipeline

Job Feed25

Outreach25

Settings

Threshold: 75%

Sources: LI + Naukri

Unscored: 0

How it works

Resume Scoring

Outreach Flow

Outreach Queue

11 jobs above 75% • review before sending

Filter: All scores

All sources

All locations

Any time

Chief Product Owner / Product Manager - ELA

Mercedes Benz - Bengaluru Naukri

78.5%

Outreach Sent

DRAFT EMAIL show

Sub: Chief Product Owner Role @ Mercedes Benz

Shortlisted Outreach Sent Replied Interview Skipped Job

Product Manager

Gocomet - Bengaluru, Thane Naukri

77.3%

Replied

DRAFT EMAIL show

Sub: Product Manager Role at Gocomet

Shortlisted Outreach Sent Replied Interview Skipped Open in Email Job

Product Manager

JP Morgan and Chase - Hyderabad Naukri

77.2%

New

Perform Outreach

Shortlisted Outreach Sent Replied Interview Skipped Job

Fig. The Outreach Queue

COIE v2.0 • Career Outreach Intelligence Engine • 10

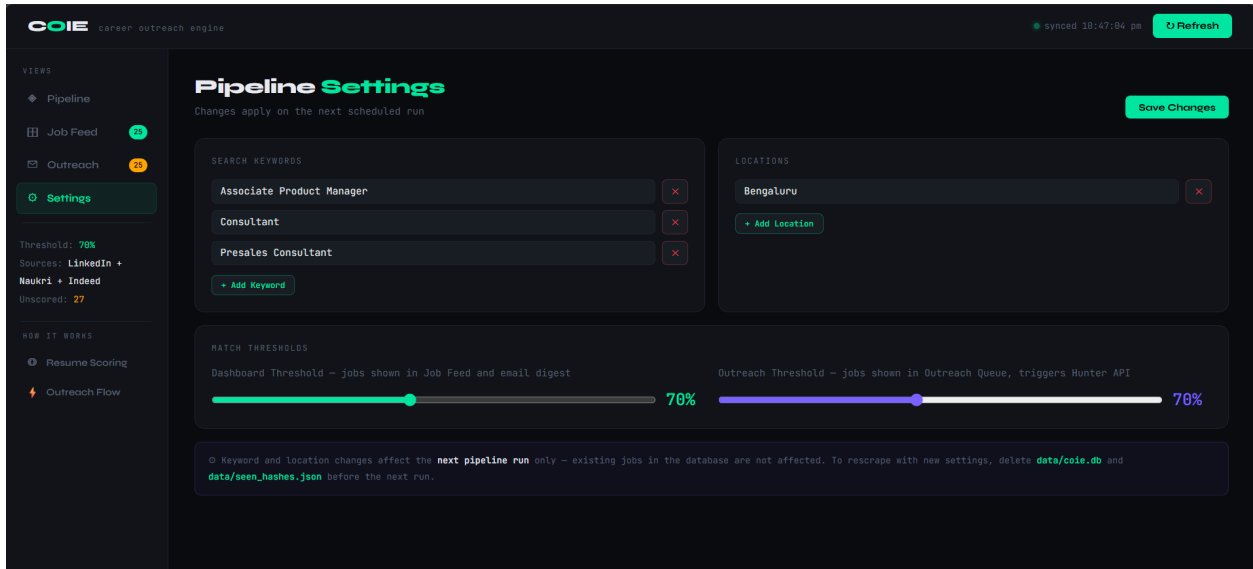


Fig. The Settings Panel

COIE v2.0 • Built and shipped Feb 2026 • Fully operational