1. Rename leads as “targeted companies”.
   1. I should have an option to enter whether the company Type is ALE, NON ALE or Good Coded
   2. Information about the co should be pulled from their website, linkeidn
   3. NEWS: any latest news indicating they are hiring or expanding in UAE?
   4. Choose location : Abu Dhabi, Dubai, Sharjah ( multiple locations to be chosen)
   5. Status: New, Contacted, Response Revd., Converted, Declined ( With Remarks)
   6. Column view: company, location, type, status, QScore, created
2. Create One more menu as “HR leads”.
   1. Each lead should be tagged under a company name
   2. Store max possible information about the lead, such as Name, Designation, Linkedin URL, company, location, role, mobile number, Email Address. ( I am not sure, do we need gender and nationality info)
   3. Leads should have information pulled from their linkedin profile, if they post any infor about their company or hiring information, so that it will be east to pitch for conversion
   4. Lead Status: New, Contacted, Response rcvd, Follow-up 1 stage, F-Up 2 stage, F-Up 3 stage, F-up 4 stage, converted, declined. (with remarks)
   5. Column view: name, company, email address, status, created (u can give me suggestions for other columns)
3. Another menu “News”: AI bot should crawl the web, linkedin for finding useful companies and leads
4. Enrichment: suggested contacts should be real time information pulled from linkedin with anticipated working email address
5. Messages: Leads should be sent an outreach (linkedin or email) ( Different templates to be created, for each campaign each template), after 1st email /msg sent, regular follow-up emails/msgs should be auto generated as per the marketing protocol until it is converted or declined. Status of the companies and leads to be updated accordingly. Email/msgs templates to be created using LLM, I can give either the campaign info as text form or upload a PDF, LLM should design nice email accordingly and shared once the template types are approved by me, several email templates to be created and approved for each stage.  
     
   The main concept is identifying the lead by the AI bot, this is very crucial step, we can work around on this with several practices.

**UPR – Product Flow Spec (v0.1)**

*Date: Sep 13, 2025 | Owner: Sivakumar C*

**0) Scope & Goals**

**Goal:** Automate discovery of premium employers entering/expanding in the UAE, enrich with decision-maker contacts, and run multi‑step, multi‑channel outreach until **Converted** or **Declined**. Maintain full auditability and metrics.

**Key Modules:**

1. Targeted Companies (formerly “Leads”)
2. HR Leads (people/contacts, linked to companies)
3. News Radar (signal discovery)
4. Enrichment (profile + contacts + emails)
5. Messages (LLM‑generated templates, sequenced outreach)
6. Auth/Admin + Reporting

**1) Targeted Companies**

**Rename**: “Leads” → **Targeted Companies**

**Data Model: targeted\_companies**

* id (uuid, pk)
* name (text, required, unique per tenant)
* type (enum: ALE, NON\_ALE, Good Coded)
* locations (string[]; values from enum: Abu Dhabi, Dubai, Sharjah; multi‑select)
* website\_url (text, nullable)
* linkedin\_url (text, nullable)
* about\_blurb (text, short summary scraped)
* news\_signals (jsonb) — array of latest relevant headlines with source/date/url
* hiring\_signal (bool, derived from news/linkedin; nullable)
* qscore (int, 0–100; see **QScore** below)
* status (enum: New, Contacted, Response Received, Converted, Declined)
* status\_remarks (text)
* created\_at (timestamptz, default now)
* updated\_at (timestamptz)

**QScore (Quality Score) – v1 heuristic**

* Base from market tier & salary signals (premium sector +10..20)
* Hiring/expansion news in UAE last 90d (+15)
* Fresh UAE job posts (last 30d) (+10)
* Decision‑maker detected with validated email (+15)
* Multi‑location in UAE (+5)
* Negative signals (layoffs/legal) (−20)

**UI – Column View (list)**

* **company**, **location(s)** (badges), **type**, **status** (chip), **QScore**, **created**

**Create/Edit Form**

* Company Name, Type (ALE/NON ALE/Good Coded), Locations (multi), Website, LinkedIn
* **Fetch Profile** button → pulls (about, logo, sector) from site/LinkedIn
* **Check News** → latest 5 headlines + hiring flag
* Save → creates/updates record; recompute QScore

**Status Machine & Notes**

* Transitions: New ↔ Contacted ↔ Response Received → Converted or Declined
* All changes logged in activity\_log with actor/time/remarks

**2) HR Leads**

People attached to companies.

**Data Model: hr\_leads**

* id (uuid, pk)
* company\_id (fk → targeted\_companies.id, required)
* name (text)
* designation (text)
* linkedin\_url (text)
* location (enum: Abu Dhabi, Dubai, Sharjah, Other)
* role\_keywords (text[]) — e.g., HR Director, TA Manager, Payroll, Finance
* mobile (text, nullable, masked in UI)
* email (text, nullable) — validation status stored in email\_status
* email\_status (enum: unknown, patterned, guessed, validated, bounced)
* gender (text, nullable, optional; **off by default**)
* nationality (text, nullable, optional; **off by default**)
* lead\_status (enum: New, Contacted, Response Received, Follow‑up 1, Follow‑up 2, Follow‑up 3, Follow‑up 4, Converted, Declined)
* status\_remarks (text)
* signal\_feed (jsonb) — latest LinkedIn posts by/contact mentions of hiring
* created\_at, updated\_at

**UI – Column View (list)**

* **name**, **company**, **email** (chip with status), **status**, **created**  
  Suggested extras: designation, location, last\_contacted\_at

**Capture & Sync**

* Create Lead → attach to selected company
* **Pull from LinkedIn**: name/title/profile URL, recent post snippets with hiring signals  
  *Note: Use official APIs/compliant providers (Sales Navigator/LinkedIn Marketing Dev Platform) or third‑party enrichment (Apollo, PDL); avoid ToS‑violating scraping.*
* Email discovery: domain pattern + MX check + verification (e.g., NeverBounce/ZB)

**3) News Radar**

Automated discovery of signals about companies entering/expanding in the UAE.

**Data Model: news\_items**

* id (uuid, pk)
* company\_name (text)
* company\_id (fk nullable; link if match found)
* title, summary, url, source, published\_at
* tags (text[]) — hiring, expansion, new office, funding, leadership hire
* score (int)
* ingested\_at

**Sources & Pipelines**

* News APIs (GDELT, NewsAPI, Google Alerts RSS)
* Company newsroom RSS, press releases
* Job boards (LinkedIn Jobs, Indeed, Naukrigulf) — count & recency
* LinkedIn company posts (official/compliant access)

**UI**

* **News menu**: feed with filters (time range, tag, location).
* Actions: “Add as Targeted Company” (pre‑fill) or “Attach to Company”.

**4) Enrichment**

Real‑time contact suggestion + email guess/verification.

**Data Model: enrichment\_jobs**

* id, input (url/name/domain), status (queued/running/done/failed)
* suggested\_contacts (jsonb[]) — name, title, linkedin\_url, email\_guess, score
* provider (enum: Apollo/Clearbit/PDL/Hybrid), ran\_by, created\_at

**Flow**

1. User submits company URL/LinkedIn or selects a Targeted Company.
2. System fetches domain, industry, size; ranks target functions (HR/TA/Payroll/Finance).
3. Provider queries → returns contacts with score.
4. Email pattern inference, verify → mark email\_status.
5. User selects contacts → “Save as HR Lead(s)”.

**5) Messages (Sequenced Outreach)**

Centralized outreach engine for Email + LinkedIn; WhatsApp later.

**Data Model**

* campaigns — id, name, objective, target (company/lead filters), channel(s)
* templates — id, campaign\_id, stage (1..N), channel (email/linkedin),  
  subject, body\_md, variables (json schema), approved (bool), tone
* messages — id, lead\_id, company\_id, campaign\_id, stage, channel,  
  rendered\_subject, rendered\_body, status (queued/sent/delivered/opened/replied/bounced), provider\_msg\_id, sent\_at
* followups — id, campaign\_id, stage, wait\_days, condition (no\_reply/opened\_no\_reply)

**Flow**

1. **Template Design**: LLM drafts from campaign text/PDF → you approve per stage.
2. **Send Stage 1** to selected leads via provider (email or LinkedIn InMail connector).
3. Auto‑scheduler checks conditions → sends Follow‑up 1..4 until Converted/Declined.
4. Replies ingested (email webhook) → auto‑update lead/company status to **Response Received**.
5. Manual overrides always allowed; all events logged.

**Channels**

* **Email**: SMTP provider (e.g., SendGrid/Resend) with domain auth & tracking.
* **LinkedIn**: Use official InMail APIs/automation partners permitted by ToS.
* **WhatsApp** (later): Meta Business API; opt‑in & template approval required.

**Personalization Variables**

* {lead.name}, {lead.title}, {company.name}, {company.hiring\_signal}, {qscore}, {recent\_news.title}, {salary\_band\_hint}

**6) Auth, Roles, Admin**

* **Now**: Simple login with Admin token enforcement for sensitive routes.
* **Later**: Roles: Admin, Operator, Viewer; audit trail across state changes.

**7) API Endpoints (Draft)**

**Companies**

* GET /api/companies?search=&type=&status=&location=&sort=&page=
* POST /api/companies (name, type, locations, website, linkedin)
* GET /api/companies/:id
* PATCH /api/companies/:id (status, remarks, fields)
* POST /api/companies/:id/fetch-profile (scrape site/LinkedIn summary)
* GET /api/companies/:id/news → from news\_items
* POST /api/companies/:id/recompute-qscore

**HR Leads**

* GET /api/leads?company\_id=&status=&search=&page=
* POST /api/leads
* PATCH /api/leads/:id
* POST /api/leads/:id/enrich (refresh signal feed/email verify)

**News**

* GET /api/news?tag=&days=&q=&page=
* POST /api/news/ingest (admin)

**Enrichment**

* POST /api/enrich (input: url/name/domain)
* GET /api/enrich/:job\_id

**Messages/Campaigns**

* GET /api/campaigns
* POST /api/campaigns
* POST /api/templates (stage/channel)
* POST /api/send (campaign\_id, filter)
* Webhooks: /webhooks/email, /webhooks/linkedin

**8) UI Changes**

**Menus**

* **Dashboard** | **Targeted Companies** | **HR Leads** | **News** | **Enrichment** | **Messages** | **Admin**

**Targeted Companies Page**

* Filters: Type, Status, Location, QScore range, Date range
* Table columns: company | location(s) | type | status | QScore | created
* Row actions: View, Edit, Enrich, Open News, Create Campaign

**HR Leads Page**

* Filters: Status, Company, Email status, Location
* Table columns: name | company | designation | email (status chip) | status | last\_contacted | created
* Row actions: View, Update, Send Msg, Add to Campaign

**News Page**

* Feed with tag/location filters; action buttons → Add/Attach

**Enrichment Page**

* Input: URL/domain or choose existing company; display suggested contacts with scores; bulk save

**Messages Page**

* Campaigns list; Template editor with approval; Outbox with statuses; Follow‑up schedule view

**9) Providers & Compliance (Non‑Negotiables)**

* Avoid ToS‑violating scraping; prefer **official APIs or licensed data partners** (Apollo, Clearbit, PDL, PeopleGPT vendors).
* Email verification via NeverBounce/ZeroBounce.
* Respect privacy: mask mobile by default; disable sensitive fields (gender/nationality) unless explicitly enabled.
* Opt‑out handling & suppression lists for email compliance.

**10) Metrics & Reporting**

* Dashboard KPIs: Targeted Companies added (by week), Leads added, Emails sent, Open/Reply rates, Conversions
* Funnel per campaign: Sent → Opened → Replied → Meetings → Converted
* SLA widgets: aging by status, follow‑up due today

**11) Implementation Plan (2‑Week Sprint Draft)**

**Sprint 1**

1. DB migrations for new enums/tables (companies/leads/news/templates/messages)
2. UI rename & table columns + filters (Companies & HR Leads)
3. Enrichment job skeleton + provider adapter interface
4. Basic email delivery via SendGrid/Resend; outbound webhook ingestion

**Sprint 2**  
5. News ingest pipeline (RSS + NewsAPI), attach to companies  
6. QScore computation service + UI badges  
7. Campaign + Template editor (LLM draft + approval flag)  
8. Sequencer for follow‑ups (cron worker) + status updates

**12) Open Questions for You**

1. Confirm Good Coded meaning (government‑coded / special‑coded?)
2. Minimum data partner to start (Apollo vs Clearbit vs PDL)?
3. Outreach channel to enable first (Email vs LinkedIn InMail)?
4. Follow‑up cadence defaults (e.g., Day 0, +3, +7, +14, +28)?
5. Sensitive fields (gender/nationality) — keep disabled by default?

**13) Risks & Mitigations**

* **API access/limits** → start with one provider; cache & backoff
* **Deliverability** → domain warm‑up, DKIM/SPF/DMARC, throttling, ramp plans
* **Compliance** → explicit unsubscribe, CRM suppression, lawful basis notes
* **Data quality** → multi‑source enrichment; verification gates before send

**14) Ready‑Next Checklist**

* Approve data models & statuses
* Confirm providers (enrichment + email)
* Approve UI menu/columns & filters
* Approve follow‑up cadence & template workflow
* Green‑light Sprint 1 scope

UPR/

├─ server.js # Express entry: mounts API routes, auth, health, serves SPA

├─ Dockerfile # Multi-stage: build dashboard, run node server.js

├─ package.json / package-lock.json # Scripts, deps

├─ dashboard/ # Vite + React frontend

│ ├─ index.html

│ └─ src/

│ ├─ pages/

│ │ ├─ DashboardHome.jsx # KPIs (to wire: counts, funnel, SLAs)

│ │ ├─ CompaniesPage.jsx # Targeted Companies list (type/status/location/QScore)

│ │ ├─ HRLeadsPage.jsx # HR leads list (status, email chip, last\_contacted)

│ │ └─ EnrichmentPage.jsx # Always-on LLM badge + contacts table + Save-as-Lead

│ ├─ components/

│ │ └─ Sidebar.jsx # Menu: Dashboard, Targeted Companies, HR Leads, News, Enrichment, Messages, Admin

│ └─ utils/

│ └─ auth.ts # authFetch: attaches JWT; JSON helpers

│

└─ server/ # Backend modules

├─ routes/

│ ├─ enrich.js # POST /api/enrich → aiEnrichFromInput → email pattern/verify → normalized \_meta

│ ├─ companies.js # /api/companies CRUD, fetch-profile, news, recompute-qscore

│ ├─ hr-leads.js # /api/hr-leads CRUD, from-enrichment, verify/enrich

│ ├─ news.js # /api/news feed & ingest

│ └─ auth.js # /api/auth/login, /api/auth/verify (JWT)

└─ utils/

├─ ai.js # aiEnrichFromInput (LLM/provider shaping)

├─ jwt.js # signJwt, verifyJwt

├─ emailPatterns.js # detectEmailPattern, generateEmail, generateCandidates

├─ emailVerify.js # verifyEmail (NeverBounce/ZB or SMTP probe)

└─ patternCache.js # getDomainPattern/setDomainPattern (per-domain cache)

**Responsibilities at a glance**

* **server.js**: wires routes, CORS/body parsing, GET /health & /\_\_diag, static serve of dashboard/dist, SPA fallback (no-store for index.html).
* **Routes**:
  + /api/enrich: Turns input (domain/URL/brief) → { company, contacts[], outreachDraft, \_meta:{model,duration\_ms} }.
  + /api/companies: Targeted Companies (type ALE/NON\_ALE/Good Coded, locations, status, qscore, news).
  + /api/hr-leads: People tied to companies (email status, follow-up stages, remarks).
  + /api/news: News radar (hiring/expansion tags; attach/add company).
  + /api/auth: Username/password → JWT; verify endpoint for client boot.
* **Frontend pages**:
  + **Enrichment**: Always-visible LLM status pill; table with email chips; Save-as-Lead button.
  + **Companies**: column view: company | locations | type | status | QScore | created; filters.
  + **HR Leads**: name | company | designation | email (status) | status | last\_contacted | created.
  + **Dashboard**: counts + funnel (Queued task to wire real metrics).

**Environment & knobs (Render)**

* UPR\_ADMIN\_USER, UPR\_ADMIN\_PASS — login
* JWT\_SECRET — JWT signing
* DATABASE\_URL — Postgres (pooler URL, sslmode=require)
* OPENAI\_API\_KEY (+ OPENAI\_MODEL optional) — LLM
* SMTP\_VERIFY\_ENABLED=true|false, SMTP\_VERIFY\_MAX=3
* SOURCING\_WORKER\_ENABLED=false (worker intentionally off for now)

**Data model (conceptual)**

* **targeted\_companies**: name, type (ALE/NON\_ALE/Good Coded), locations[], website/linkedin, about\_blurb, news\_signals[], hiring\_signal, qscore, status, remarks, timestamps.
* **hr\_leads**: company\_id, name, designation, linkedin\_url, location, role\_keywords[], email, email\_status (unknown/patterned/validated/bounced), lead\_status (New → Followups → Converted/Declined), signal\_feed[], status\_remarks, timestamps.
* **news\_items**: company\_name/id, title, summary, url, source, tags, published\_at, score.

**Request flows (quick)**

* **Enrich**: UI → POST /api/enrich {input} → LLM + contact provider → email pattern/verify → normalized \_meta → UI renders badge/table → Save-as-Lead → POST /api/hr-leads/from-enrichment.
* **Auth**: Login → JWT stored → all API calls via authFetch (Bearer token).

**Build & deploy**

* Docker multi-stage: build React app inside /dashboard, copy dist/, run node server.js.
* Render health: /health (OK), /\_\_diag (db\_ok etc., if implemented).