# 📦 Project Summary: **Tradersecho — Twitter Sentiment App**

## 🔑 Important details

* **GitHub repo**: <https://github.com/Mirrormoney/Tradersecho>
* **Local path**: C:\Users\mai-s\Desktop\TradersEcho
* **Backend framework**: FastAPI + SQLAlchemy + Alembic
* **Frontend**: React (Vite, Tailwind-like styling)
* **DB**: PostgreSQL (driver: psycopg[binary])
* **Seeded users**:
  + admin@example.com / admin123 (pro=True)
  + user@example.com / user123 (pro=False)
* **Admin token** (for promotions): change-admin-token (from .env)

## 🛠️ What we already built & fixed

* ✅ **API backend** with:
  + /api/free/daily → DB-backed yesterday totals (with date filters + pagination).
  + /api/pro/snapshot → realtime snapshot (last 5 min aggregated vs baseline).
  + /ws/realtime → live WebSocket (Pro only).
  + Auth: signup/login + JWT tokens.
  + /api/me returns current user + pro flag.
  + /api/admin/make-pro to promote users temporarily.
* ✅ **DB schema**:
  + users
  + mention\_minutes (per-minute cashtag mentions, pos/neg/neu)
  + daily\_rollups (totals per ticker/day, interest, zscore)
  + baselines (mean/std for normalization)
* ✅ **Collector** (collector.py):
  + python collector.py backfill --days 14 → realistic volumes (~tens of thousands/day).
  + python collector.py live → inserts new minute rows continuously.
* ✅ **Jobs** (jobs.py):
  + python jobs.py baselines --window 30 → recompute means/stds.
  + python jobs.py rollup --date YYYY-MM-DD → recompute a day from minute data.
* ✅ **Frontend GUI**:
  + Free (delayed) list (with filters).
  + Login/Signup.
  + Upgrade-to-Pro (Stripe placeholder).
  + Pro realtime stream.
* ✅ **Alembic** configured to load .env → no more DB\_URL mismatch.
* ✅ **Postgres driver** switched to psycopg[binary]==3.2.10 → no Visual Studio compile errors.
* ✅ **All-in-one release (v1.4.2)**: includes backend + frontend, fully working.

## 🔜 Plan going forward

We agreed on this sequence:

### **Option A (done)** ✅

* Switch to Postgres as default.
* Add collector + jobs.
* Add rollups, baselines, indices.
* DB-backed API endpoints.
* ✔️ Completed in v1.3 → consolidated into v1.4.2.

### **Option C (next)**

* Add **real collector adapters** (pluggable):
  + Reddit, StockTwits, maybe X (if keys available).
  + Deduplication, rate-limiting.
  + source column in DB.
* Data still flows into mention\_minutes so existing API/UI works unchanged.

### **Option B (later, polish)**

* Frontend enhancements:
  + Charts/sparklines.
  + Market overview card.
  + Export CSV.
  + Pro features: adjustable refresh, alerts.

## 🔧 How to run locally (short)

Backend:

cd C:\Users\mai-s\Desktop\TradersEcho\twitter-sentiment-app-v1.4.2\backend

.venv\Scripts\activate

pip install -r requirements.txt

alembic upgrade head

python seed.py

python collector.py backfill --days 14

python jobs.py baselines --window 30

python -m uvicorn app:app --host 127.0.0.1 --port 8000 --reload

Frontend:

cd C:\Users\mai-s\Desktop\TradersEcho\twitter-sentiment-app-v1.4.2\frontend

npm install

npm run dev

Open: <http://127.0.0.1:5173>

## 🧭 What to tell me in a new chat

If you start a fresh chat, just copy-paste this summary and say something like:

“Continue from here — we’re on Tradersecho v1.4.2. Backend is FastAPI + Postgres (psycopg binary). Frontend React. GitHub repo is <https://github.com/Mirrormoney/Tradersecho>. Local path is C:\Users\mai-s\Desktop\TradersEcho. We completed Option A, next step is Option C (collector adapters).”