אינטראקטיב ישראל

סיכום הקורס

השקעות AI

בניית דשבורד שוק

פרומפט ליצור את הממשק לבד ב-Plan Mode

Build me a working Hebrew RTL trading-news dashboard called "tradingdashboard" that aggregates posts about tech/AI/crypto tickers from X (Twitter), Reddit, RSS, and Hacker News, classifies the tone of each item in Hebrew via the Claude Code CLI, and shows mention velocity per ticker over time. It must be a real working product I use daily AND a polished live demo for my Hebrew AI Agent School workshop. Israeli regulatory framing: educational tool only, no investment advice, no buy/sell recommendations, no price predictions.

Audience and language

- Israeli users. UI chrome is 100% Hebrew (every label, button, error, empty state, tooltip, modal, page title, alt-text, aria-label, page metadata). Source content can stay in its native language (mostly English).
- All visible strings centralized in lib/copy.ts. Zero inline Hebrew in JSX.
- Hebrew style: plain everyday Israeli, never formal/literary. "מכירים את" not "ידנית". "יודעים את" not "ביד". Punctuation AFTER text.
- Dates via Intl.DateTimeFormat('he-IL', { timeZone: 'Asia/Jerusalem' }). Numbers via Intl.NumberFormat('he-IL'). Wrap both in a lib/format.ts helper.

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

Data sources (exactly 4)
- Twitter: Apify actor apidojo/tweet-scraper, ~20 fintwit + AI handles.
- Reddit: Apify actor trudax/reddit-scraper-lite, subs investing / stocks / wallstreetbets / CryptoCurrency / MachineLearning.
- RSS: rss-parser. Do NOT use Bloomberg or Reuters (dead public RSS). Use Google News search RSS, Yahoo Finance per-ticker, CNBC, MarketWatch.
- Hacker News: Algolia HN Search API (free, no auth). 3 parallel keyword buckets: stocks / crypto / AI-tech.
- Only one secret: APIFY_API_TOKEN. NO Anthropic API key, NO twitterapi.io. Sentiment runs via `claude` CLI subprocess so the user's subscription pays, not an API key.

Tech stack
- Next.js 16 + App Router + TypeScript strict
- SQLite via better-sqlite3 at db/tradingdashboard.db
- Tailwind v4 + recharts + Lucide icons
- Local-only on 127.0.0.1:3003. No auth, no cloud deploy.

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

Visual design
- TradingView terminal aesthetic, NOT bento, NOT Apple-style.
- Palette: page #131722, panel #1E222D, border #2A2E39, text #D1D4DC, muted #787B86, accent #2962FF, amber CTA #F59E0B, cyan #26C6DA.
- Layout: 32px top bar (brand + ticker tape + sync pill), 28px disclaimer strip, collapsible left rail (48 <-> 200), main area, collapsible right watchlist (260, conditional per route), Cmd+B bottom panel.
- Font: Noto Sans Hebrew for UI, JetBrains Mono with tabular-nums for numbers and tickers.
- BANNED CLASSES: scale-*, rounded-2xl, shadow-*. Terminal density only. Hover = subtle bg shift 100ms, no scale.
- Sentiment chips are GRAY only (#363A45 bg, #D1D4DC text). NO red/green, NO arrows. Labels: "טון חיובי / טון שלילי / טון ניטרלי" (tone descriptive, never bullish/bearish - that reads as buy/sell signal to a regulator).

סיכום הקורס

השקעות AI

Hebrew-first guarantees (NON-NEGOTIABLE)
- <html lang="he" dir="rtl"> in root layout.
- Sentiment analyzer's prompt requests Hebrew 1-sentence summary_he per (item, ticker), classifying tone of TEXT (not investment direction).
- Disclaimer strip pinned to every page + first-load modal dismissed to localStorage + full Hebrew /terms page.

Build orchestration: 5 sequential waves of parallel subagents
Each wave's agents work in parallel with strict file-allowlists so they can't race on shared files. Each wave has a real validator script that must exit 0 before the next wave starts.

- Wave 0 (1 agent, sequential): scaffold project, write lib/db.ts with FULL helper API (insertItems, getExistingExternalIds, tagTickers, getUnscoredItems, upsertSentiment, upsertTrendSnapshot, normalizeEngagement), write lib/types.ts as a LOCKED contract, lib/tickers.ts + lib/ticker-tagger.ts, lib/copy.ts (Hebrew strings), lib/claude-bridge.ts with semaphore (max 3 concurrent) + 60s rate-limit backoff + daily cap 500 calls, copy design-system folder, swap font + palette. Wave 1 agents are FORBIDDEN from npm install and from editing shared files.

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

- Wave 1 (4 parallel agents, one per source): each owns exactly one file under lib/sources/, implements the Ingestor interface from lib/types.ts, normalizes to NormalizedItem schema, dedupes on external_id, 15-min disk cache. Each writes a smoke test.

- Wave 1 validator: each source has >= 5 items in last 24h, all external_ids match ^(twitter|reddit|rss|hn):.+, no null published_at or url, HEAD-request sample URLs per source (accept 403 for x.com - bot-blocked but URL shape valid), at least one item per source has a tagged ticker. Exit non-zero on any failure.

- Wave 1 validator: each source has >= 5 items in last 24h, all external_ids match ^(twitter|reddit|rss|hn):.+, no null published_at or url, HEAD-request sample URLs per source (accept 403 for x.com - bot-blocked but URL shape valid), at least one item per source has a tagged ticker. Exit non-zero on any failure.

- Wave 2 (2 parallel agents): Agent E = sentiment, hard-batches 20 items per Claude CLI call, JSON array out, validates label/score/confidence ranges, hard daily cap 500. Agent F = trends, pure SQL aggregations, no LLM, UPSERT trend_snapshots, spike detection at velocity > 2.0 and mention_count >= 3.
- Wave 2 validator: >= 80% of (item, ticker) pairs scored, all scores in [-1,1], all sentiment summaries contain Hebrew chars, all enums canonical, trend_snapshots covers every tagged ticker.

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

- Wave 3 (1 agent, biggest): TradingView terminal UI per spec above. Build 5 pages (overview / feed / trends / sources / terms), 5 API routes (/api/sync POST + /api/sync/status GET polling NOT SSE + 3 metrics routes), ~14 primitives including TerminalCard, TickerTape with pause-on-hover and prefers-reduced-motion, SentimentChip, DisclaimerStrip + Modal, SourceBadge, FeedRow at 28px dense rows, recharts with contentStyle direction:'rtl' on every Tooltip. Pre-delivery checklist as a HARD gate: no emoji icons, no banned classes, all dates via formatDate, all numbers via formatNumber, zero inline Hebrew in JSX except terms/page.tsx prose, npm run build clean, 8/8 HTTP routes return 200.

- Wave 4 (1 agent): server-side 5-min sync rate gate that returns 429 with retry_after_seconds + ?force=1 bypass, prewarm cron script that refreshes RSS+HN+Twitter every 5 min (NOT Reddit, Apify cold-start hurts demo), bilingual README with workshop demo arc, .env.example with only the keys (no values).

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

- Wave 3 (1 agent, biggest): TradingView terminal UI per spec above. Build 5 pages (overview / feed / trends / sources / terms), 5 API routes (/api/sync POST + /api/sync/status GET polling NOT SSE + 3 metrics routes), ~14 primitives including TerminalCard, TickerTape with pause-on-hover and prefers-reduced-motion, SentimentChip, DisclaimerStrip + Modal, SourceBadge, FeedRow at 28px dense rows, recharts with contentStyle direction:'rtl' on every Tooltip. Pre-delivery checklist as a HARD gate: no emoji icons, no banned classes, all dates via formatDate, all numbers via formatNumber, zero inline Hebrew in JSX except terms/page.tsx prose, npm run build clean, 8/8 HTTP routes return 200.

- Wave 4 (1 agent): server-side 5-min sync rate gate that returns 429 with retry_after_seconds + ?force=1 bypass, prewarm cron script that refreshes RSS+HN+Twitter every 5 min (NOT Reddit, Apify cold-start hurts demo), bilingual README with workshop demo arc, .env.example with only the keys (no values).

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

Plan-time discipline (before any code)
Enter plan mode. Use Explore subagents in parallel to map existing patterns. Then write a detailed plan to a plan file. Then spawn 5 reviewer subagents in parallel to stress-test the plan from different angles (wave dependency feasibility, schema correctness, external API risk, UI lift realism, workshop demo + compliance viability). Revise the plan based on the 5 reviews before exiting plan mode. Use the ui-ux-pro-max skill (--design-system --persist) to generate the design tokens before Wave 0.

What NOT to do
- Don't show prices anywhere. We track volume + tone, not prices.
- Don't use "bullish" / "bearish" labels (regulatory risk in Hebrew audience). Use "positive/negative/neutral" with Hebrew "טון" prefix.
- Don't use SSE for sync status streaming - Next App Router buffers it in dev mode and breaks the live demo. Use polling /api/sync/status every 800ms.
- Don't use named-param :bind binding with integer-division SQL in better-sqlite3 - it promotes to REAL and breaks bucketing. Use positional ? or inline the bucket constant.
- Don't commit .env.local, db/, .next/. Verify with git ls-files --others --cached --exclude-standard before first push.
- Don't use em-dash, en-dash, or fancy quotes. Plain hyphen-minus only.
- Don't use LLM cliche words: delve, leverage, robust, seamless, tapestry, navigate, "in today's world".
- Don't deploy to cloud. Local-only on 127.0.0.1.

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

Deliverables
- Working project at ~/path/tradingdashboard/
- Bilingual README (Hebrew + English) with 7-step quick start each
- .env.example showing keys only, no values
- 5-wave validator scripts that prove each gate
- Public GitHub repo (only after verifying no secrets in staged files)
- Final report with: per-source item counts, sentiment coverage %, build status, all 8 HTTP route status codes, total cost estimate per workshop

Build it.

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

איך לעקוב אחרי הכסף הגדול

פקודה לשים בטרמינל / Powershell להורדה מלאה

```
cd sec-scanner
cp install/.env.example .env
# edit .env, paste your FMP_API_KEY and your name+email as EDGAR_IDENTITY
bash scripts/setup.sh
source .env
bash scripts/verify.sh
```

פקודות ויכולות שניתן לבצע עם הסקילים

/morning-brief	הפקודה הראשית של הבוקר. סורקת את כל ה-watchlist, מצליבה דיווחי SEC + סיגנלים, ומחזירה רשימה מדורגת של 5 ה-tickers הכי בולטים היום. תריץ אותה כל בוקר לפני שתפתח את השוק.
/insider-watch	מושכת את כל דיווחי Form 4 האחרונים על המנהלים הבכירים של הטיקרים ב-watchlist שלך. רואה מי קונה ומי מוכר, באיזו כמות ובאיזה מחיר. מסמנת קניות מעל מיליון דולר ומכירות מעל 5 מיליון.
/flow-track	מקבלת שם של גורו (לדוגמה berkshire_hathaway), שולפת את ה-13F האחרון שלו, ומראה לך מה הוא מחזיק, באיזה שווי ובאיזה אחוזים מהתיק. בנוסף נותנת לך סיגנל משוקלל מה-Institutional Flow Tracker.

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

/portfolio-diff	הפקודה הראשית של הבוקר. סורקת את כל ה-watchlist, מצליבה דיווחי SEC + סיגנלים, ומחזירה רשימה מדורגת של 5 ה-tickers הכי בולטים היום. תריץ אותה כל בוקר לפני שתפתח את השוק.
/thirteen-d	מושכת את כל דיווחי Form 4 האחרונים על המנהלים הבכירים של הטיקרים ב-watchlist שלך. רואה מי קונה ומי מוכר, באיזו כמות ובאיזה מחיר. מסמנת קניות מעל מיליון דולר ומכירות מעל 5 מיליון.
/exec-comp	מקבלת שם של גורו (לדוגמה berkshire_hathaway), שולפת את ה-13F האחרון שלו, ומראה לך מה הוא מחזיק, באיזה שווי ובאיזה אחוזים מהתיק. בנוסף נותנת לך סיגנל משוקלל מה-Institutional Flow Tracker.
/weekly-newsletter	רץ פעם בשבוע (ביום שישי), אוסף את כל מה שקרה ב-7 הימים האחרונים, ומפיק 5-10 כותרות בעברית לניוזלטר שאפשר לשלוח באימייל.

דוגמאות לאיך להשתמש בסקילים

/morning-brief
/insider-watch
/flow-track berkshire_hathaway
/flow-track michael_burry
/flow-track bill_ackman
/portfolio-diff berkshire_hathaway 2025Q4 2026Q1
/thirteen-d TSLA
/thirteen-d NVDA
/exec-comp NVDA
/exec-comp META
/weekly-newsletter

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

איך הפלואו של הפקודות זורם

פקודה → JSON ב-render.py → data/snapshots/ מייצר → דשבורד / אקסל / PDF / ניוזלטר

כל פקודה אטומית. אפשר להריץ שוב ושוב, היא תכתוב מחדש את הקובץ שלה בלי לפגוע באחרים.

פרומפט ליצור את הממשק לבד ב-Plan Mode

Build me a local Claude Code system called [SEC Scanner] that tracks [institutional + insider activity on US-listed stocks] using only public data. Target audience: [Israeli retail traders, non-tech, Hebrew RTL UI].

DATA SOURCES (all free tier):
1. EdgarTools remote MCP for SEC EDGAR (13F, Form 4, Schedule 13D/G, DEF 14A). Register via:
   claude mcp add edgar_tools --transport http https://app.edgar.tools/mcp/
2. Institutional Flow Tracker Claude Code skill (uses FMP API for tier-weighted superinvestor signals). Install from github.com/tradermonty/claude-trading-skills.
3. Optional: GuruFocus MCP (https://api.gurufocus.com/mcp, Bearer token).

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

ARCHITECTURE:
- Slash commands in .claude/commands/ orchestrate MCP calls and write normalized JSON to data/snapshots/ matching a SCHEMAS.md contract.
- Python renderers in outputs/ consume that JSON to produce:
  - Hebrew RTL HTML dashboard (vanilla JS, no build step)
  - openpyxl Excel per-guru sheets
  - PDF reports via Chrome headless (no system-lib deps)
  - jinja2 markdown newsletter
  - colorama terminal screener
- Pure-Python live-pull.py bypasses MCPs for cron/CI usage (hits FMP REST + SEC EDGAR directly).
- 13F XML parser (scripts/parse-13f.py) fetches informationTable XML, parses positions, resolves CUSIPs to tickers via OpenFIGI free API.
- Alerts deriver (scripts/derive-alerts.py) runs 6 rules over snapshots and produces alerts-{date}.json that the dashboard picks up.
- Scheduler: GitHub Actions workflow (weekday cron) + macOS launchd installer for "set it and forget it" daily runs.

SLASH COMMANDS:
- /morning-brief - daily orchestrator, top-5 ranked tickers
- /insider-watch - Form 4 for watchlist tickers
- /flow-track {guru_id} - latest 13F + tier signals for one guru
- /portfolio-diff {guru} {from_q} {to_q} - quarter-over-quarter diff
- /thirteen-d {ticker} - >5% stakeholders (Schedule 13D vs 13G)
- /exec-comp {ticker} - DEF 14A exec compensation table
- /weekly-newsletter - 7-day rollup into Hebrew newsletter items

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

CONSTRAINTS:
- Hebrew strings: casual 25-year-old WhatsApp style, never literary. Plain hyphen only, never em/en dash.
- All keys via .env (gitignored). Sample .env.example committed.
- EDGAR_IDENTITY env var = "Name email" per SEC fair-access policy.
- Free tier limits hardcoded in code comments: FMP 250/day page=0 only, EdgarTools hosted 100/day.
- Plain JSON files as persistence, no DB. Atomic writes (.tmp -> rename).
- No emoji icons, use Lucide-style SVG.
- WCAG AAA dark mode default, light toggle, Heebo + JetBrains Mono.
- Legal disclaimer at top of every visible output: "public information aggregator, not investment advice" - lawyer review pre-publish.

DELIVERABLES:
- Working `bash scripts/setup.sh` that installs both MCPs idempotently
- Working `bash scripts/verify.sh` with 5 pass/fail checks
- Bilingual README (English + Hebrew) walking a stranger from clone to first dashboard refresh in 7 steps
- MIT LICENSE, GitHub Actions CI (syntax + JSON validity + mock renders), issue + PR templates, dashboard screenshot in README
- Initial public push to github.com/{user}/{repo} with `gh repo create`

PROCESS:
- Spawn parallel build agents for independent layers (install / commands / renderers / scheduler / parser). Define JSON schemas BEFORE building so the layers integrate without later patching.
- After each build round, run the full pipeline against mock fixtures and against one real ticker (e.g. Berkshire 13F totalling ~$263B) to verify correctness end-to-end.
  Start by creating SCHEMAS.md with the JSON contract, then fan out.

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

CONSTRAINTS:

- Hebrew strings: casual 25-year-old WhatsApp style, never literary. Plain hyphen only, never em/en dash.
- All keys via .env (gitignored). Sample .env.example committed.
- EDGAR_IDENTITY env var = "Name email" per SEC fair-access policy.
- Free tier limits hardcoded in code comments: FMP 250/day page=0 only, EdgarTools hosted 100/day.
- Plain JSON files as persistence, no DB. Atomic writes (.tmp -> rename).
- No emoji icons, use Lucide-style SVG.
- WCAG AAA dark mode default, light toggle, Heebo + JetBrains Mono.
- Legal disclaimer at top of every visible output: "public information aggregator, not investment advice" - lawyer review pre-publish.

DELIVERABLES:

- Working `bash scripts/setup.sh` that installs both MCPs idempotently
- Working `bash scripts/verify.sh` with 5 pass/fail checks
- Bilingual README (English + Hebrew) walking a stranger from clone to first dashboard refresh in 7 steps
- MIT LICENSE, GitHub Actions CI (syntax + JSON validity + mock renders), issue + PR templates, dashboard screenshot in README
- Initial public push to github.com/{user}/{repo} with `gh repo create`

אינטראקטיב ישראל

סיכום הקורס

השקעות AI

בעידן ה־AI TradingView

פקודות ויכולות שניתן לבצע עם הסקילים

קישור הורדה ישיר:

https://github.com/peleg-jpg/tradingview-mcp-pelegdror

"Look at this chart and tell me what I'm seeing." Like having an analyst look over your shoulder for 30 seconds.	/chart-analysis
"Compare these 5 tickers, which one's set up best right now?" A conversational screener.	/multi-symbol-scan
"Build me an indicator that does X." Writes Pine, compiles, fixes its own errors, repeats until clean. Kills the Pine Editor + Stack Overflow loop.	/pine-develop
"Let's paper trade March 2024 like it's live." Rewinds the chart, you call buys/sells, it tracks P&L. Practice without risk.	/replay-practice
"Tell me if this strategy is actually any good." Net profit, win rate, profit factor, max drawdown, equity curve, plus concrete recommendations like "tighten your entries" or "fix your position sizing."	/strategy-report
Same as strategy-report but runs autonomously so your main chat stays clean.	/performance-analyst (subagent)