

Demo Pump

Darkfi Coin Market-Making Demo

Project documentation — what is included and what each part is for

Generated: 2026-09-28

1. Executive summary

Demo Pump is a web-first market-making demonstration built for Darkfi Coin (ticker DRK, pair DRK/USDT). It was created from client requirements to clone ADAMANT Market-Making Software concepts, surface premium features, add simulation mode, and present everything in a modern dark UI that can go live on the web.

The product name is Demo Pump. The focus asset is Darkfi Coin. For the demo phase, paper/simulation trading is the default so stakeholders can evaluate market quality (depth, spread, volume, chart behavior) without exchange API keys.

Client goals covered

- Clone / adapt adamant-tradebot as the foundation
- Brand as Demo Pump with coin name Darkfi Coin
- Include premium-style market-making features
- Add simulation mode for safe demos
- Ship a modern attractive dark web UI
- Make it runnable live on the web
- Provide this PDF explaining the project

2. What is in the project

2.1 Next.js application (demo-pump/)

A single Next.js 15 app hosts both the frontend dashboard and the backend API routes. This keeps deployment simple for a web demo: one process, one URL.

- Landing page — brand-first intro for Demo Pump / Darkfi Coin
- Dashboard (/dashboard) — live console with charts, order book, trades, feature toggles
- Docs page (/docs) — human-readable overview + PDF download
- API routes under /api/* — status, market snapshot, feature catalog, bot control

2.2 Simulation engine (src/lib/simulation/)

An in-memory simulation engine ticks on an interval and produces realistic market microstructure for DRK/USDT: mid price, bid/ask, order book depth, candles, fills, balances, spread (bps), open orders, and simulated PnL. Feature toggles change behavior (tighter spreads with liquidity, denser books with order-book builder, smoother candles with nice-chart / volatility modules, etc.).

2.3 Trading engine copy (engine/)

The original adamant-tradebot codebase is copied into demo-pump/engine/. This is the production-oriented Node market-making bot with exchange connectors, MongoDB persistence, command system, and WebUI API. Use it when moving from simulation to live CEX market-making.

- engine/trade/ — trader, order book builder, liquidity provider, price watcher, exchange connectors
- engine/modules/ — config, DB, commands, market-making helpers

- engine/api/ — Fastify WebUI API + Socket.IO (premium-style control surface)
- engine/config.default.jsonc — all bot parameters with comments

2.4 Frontend UI

Dark trading-terminal aesthetic with teal/sky accents (not a generic purple theme). Components include stats cards, area chart (Recharts), depth order book, trade feed, control panel (simulation vs live prep), and a premium/core feature switchboard.

3. Feature catalog — what each module is for

- Trader (t) — creates buy/sell volume with configurable policies
- Order Book Builder (ob) — keeps visible depth and dynamic book structure
- Liquidity Provider (liq) — maintains liquidity and tighter spreads
- Price Watcher (pw) — keeps price within support/resistance guidance
- Balance Watcher (bw, premium) — monitors inventory drawdowns
- Anti-Gap (ag, premium) — fills gaps so the book looks healthy
- Cleaner (cl, premium) — removes noisy/outlier orders
- Ladder / Grid (ld, premium) — grid orders across a price range
- Quote Hunter (qh, premium) — seeks better quote placement
- Volatility Chart (vc, premium) — shapes candle volatility
- Volume Volatility (vv, premium) — adds realistic volume variance
- Price Maker Volume (pmv, premium) — extra volume via price-maker logic
- Sniper Bot Watcher (sbw, premium) — detects aggressive sniper behavior
- Nice Chart (nc, premium) — smoother presentation-ready chart behavior
- DEX Price Watcher (dex, premium) — reference DEX pricing guidance

4. How the pieces work together

In demo mode, the browser polls GET /api/market every ~1.5s. The simulation engine returns a MarketSnapshot. Operators pause/start/stop/reset via POST /api/control and toggle modules the same way. Simulation is the default because the client priority is a convincing Darkfi Coin demo.

When ready for live markets, configure the engine/ bot (pair, exchange keys, MongoDB) and keep the Next.js UI as the public-facing demo shell — or point live mode at the engine WebUI API. Live mode in the dashboard is labeled as preparation until exchange credentials and connectors are wired for Darkfi's listed market.

5. Running and going live on the web

Local demo

From the demo-pump folder: `npm install && npm run dev` — then open `http://localhost:3000`.
Dashboard: /dashboard. Docs: /docs. PDF: /Demo_Pump_Darkfi_Documentation.pdf

Production web deploy

- Deploy demo-pump to Vercel, Railway, Render, or any Node 20+ host
- Set build command: `npm run build` | start command: `npm start`
- Share the public URL with stakeholders as the Darkfi Demo Pump
- For real trading, run engine/ on a VPS with MongoDB and exchange API keys (never commit secrets)

6. Security & compliance notes

- Never paste exchange API keys into chat or commit them to git
- Use .env / secret managers for live credentials
- Simulation does not place real orders — safe for demos
- Live trading involves financial risk; review configs before enabling

7. Folder map

- demo-pump/src/app — pages (landing, dashboard, docs) + API routes
- demo-pump/src/components — dark UI dashboard components
- demo-pump/src/lib — config + simulation engine
- demo-pump/engine — full adamant-tradebot copy for live MM
- demo-pump/public — static assets + this PDF
- demo-pump/scripts — PDF generator and tooling

8. Branding

Product: Demo Pump. Asset: Darkfi Coin (DRK). Market shown: DRK/USDT. Tagline focus: premium market-making demo — depth, spread, volume, and chart quality — so the team can “see what Darkfi got for us” on the web.

End of document — Demo Pump / Darkfi Coin