

# Building an Automated Trading Bot with AI Support

A **regime-adaptive crypto trading bot** that runs 24/7 on its own server, trades several crypto pairs on Binance, and is controlled entirely from a phone. Built by a finance professional with very limited coding background, **working hand-in-hand with AI**.

I'm not a developer. I spend my days in the corporate finance world, not writing software code. I wanted to find out whether someone with my background could build a real, live trading bot using AI as a partner. The honest answer: **yes, and far more easily than I expected**. This is the story of what it does and how it came together.

## THE SETUP

- **Always on** — runs around the clock on a cloud server (Hetzner), surviving restarts and disconnects without manual babysitting.
- **Python engine** — a modular Python codebase connects to Binance, reads the market, and places orders automatically.
- **Phone control** — fully operated through a Telegram chat: check balances, positions and force trades from anywhere.

### How a cycle runs

1. Wake up on a timer
2. Read the market
3. Score each opportunity
4. Open or close trades with risk rules
5. Alert me on Telegram

## WHAT IT TRADES

- **Liquid pairs:** BTC, ETH, BNB, SOL, XRP, LINK, AVAX, DOGE and LTC, all against USDT, long-only, spot.
- **Reads the mood:** For each pair it checks if market is rising, falling or flat, then adapts how it behaves.
- **Capital first:** Strict position sizing, stop-losses, a daily loss limit and cooldowns keep risk under control.

## CONTROLLED FROM A TELEGRAM CHAT

The entire bot is driven from one Telegram conversation on my phone. A few of the commands:

- **/balance** — portfolio value, free funds and per-trade size
- **/trades** — open positions with live profit & loss
- **/regime** — current market mood for any pair
- **/open /close /closeall** — force trades manually
- **/status /backtest** — health check and strategy testing

## THE STRATEGY: IT READS THE MARKET

First the bot works out the **regime** — whether a pair is in a rising, falling or flat market — then changes its behaviour accordingly:

<b>Bull</b>	Rising market. Rides confirmed momentum, but only with a healthy trend and the strictest entry bar.
<b>Bear</b>	Falling market. Takes only careful oversold bounces and stays out when the downtrend is too violent.
<b>Sideways</b>	Flat market. The most conservative mode, buys clear oversold dips near the lower band only.

### How it decides

- **Reads classic indicators:** Trend, momentum, volatility and volume.
- **Scores each setup:** A trade only opens when enough signals line up, never on a single hint.
- **Adapts to the regime:** The bar to enter rises or falls depending on the market mood.
- **Risk guardrails:** 10% per trade · stop-loss + take-profit · daily loss limit · cooldowns · time stop.

### The indicators/signals used

The bot calculates these on the 4-hour timeframe, with a daily check for confirmation:

- **RSI** (momentum / overbought-oversold).
- **MACD** (momentum / trend shifts), tuned to 8/21/5, a crypto-friendly faster setting.
- **Bollinger Bands** (volatility / how stretched price is).
- **EMAs:** 50 and 200 for trend direction; a 21 EMA as an exit trigger.
- **ATR** (volatility) used to size the stop-loss and take-profit.
- **ADX** (trend strength) how strong, not just which direction.

Plus, two "smart" additions layered on top: an **RSI divergence detector** (spots when momentum is quietly turning before price does, gives a bonus to the score) and a **volume confirmation check** (a trade needs a volume spike or rising volume to qualify).

### How the three market conditions change behaviour

First, the bot decides the regime by comparing the 50 and 200 EMAs: if they're far apart and rising it's bull, far apart and falling it's bear, and close together it's sideways. Then the rules shift per regime:

**Bull** (rising market): Looks for RSI in a healthy 45–60 rising zone, positive MACD, price above the Bollinger midline. Requires a strong trend confirmed by ADX above 25, and the daily trend must also be up (50 EMA above 200 EMA on the daily). It needs the highest bar to enter, 4 of 5 conditions.

**Bear** (falling market): Hunts for oversold bounces: RSI recovering in the 35–50 range, MACD turning up, price near the lower Bollinger band. Crucially, it avoids entering when ADX is too high (above 30), a screaming downtrend is too risky to bounce-trade. Needs 3 of 5 conditions.

**Sideways** (flat market): The most conservative mode. Only buys clear oversold dips: RSI below 38 and turning up, price hugging the lower Bollinger band. No ADX filter here (trend strength is naturally low when ranging), and the EMA-based exit is switched off so it doesn't get shaken out by normal chop. Needs 3 of 5 conditions.

# Built by a non-techie, with the help of AI / LLMs

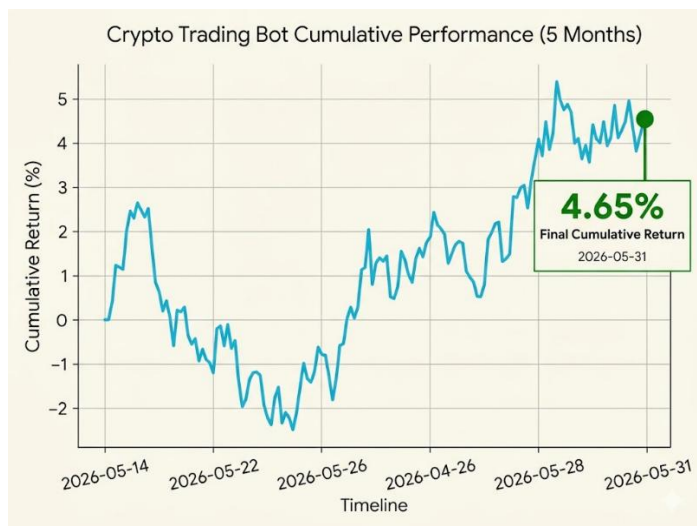
## What I brought

- The idea and the finance judgment, what to trade and what “good risk” looks like.
- The strategy logic and the rules I wanted the bot to follow.
- Direction and decisions at every fork, and a sanity check on results

## What AI did

- Wrote essentially all the code, module by module, from my plain-English asks
- Explained every concept and debugged problems I could never have solved alone
- Walked me through server setup and deployment, step by step

## Results and Takeaway



The surprise wasn't that it worked — it's **how accessible it was**. With domain knowledge on one side and AI as a patient, capable partner on the other, a finance person with little coding background can now build real, working software. Regarding the current performance, the whole point / objective was **learning to build it, not chasing returns!**

**Disclaimer:** This is a personal, educational project shared to document a learning journey. It is not financial advice, not an investment product, and not a recommendation to trade or to use any strategy described. Automated and crypto trading carry significant risk, including the loss of capital. Nothing here is a performance claim. Always do your own research.