

FREE STARTER KIT · OCTOBER 2026

Bot Builder Starter Kit

How to build a trading bot with ChatGPT or Claude **without fooling yourself**. Five tested prompts, a five-step safe workflow, and a checklist for spotting bot claims that won't hold up.

5

copy-ready prompts, each test-run in ChatGPT on 2026-10-16

5

viral AI bots rebuilt and backtested on real data since 2019-01-02

0

look-ahead bugs found in ChatGPT's scripts (but plenty of other gotchas)

RECEIPT · what's inside

Why viral bot claims fall apart	p.2
The 5-step safe build workflow	p.3
Two checks that kill most bots	p.4
5 tested prompts + their gotchas	p.5-10
Bot-claim BS-detector	p.11
Paper trading safely	p.12
Free & cheap starter tools	p.13
TOTAL	\$0.00

BUILT FOR

- Beginners curious about building a bot with ChatGPT or Claude
- Side-hustlers who saw a viral "my AI bot made 300%" post and want to check it
- Anyone who wants to learn on **paper money** before risking real money

YOU'LL NEED

A free chatbot account, free Python, and about an hour. Coding experience helps but is not required; patience is.

Data as of 2026-10-17 (daily bars). Every number in this kit comes from the Bot Receipts engine or our logged ChatGPT test. No screenshots, just receipts.

Why most viral bot claims don't hold up

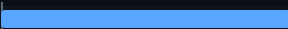
A viral bot post is a screenshot plus a story. It almost never tells you the exact rules, the data, the fees, or what the bot was compared against. When we rebuild these bots from their own descriptions and run them on real market data, the story usually changes.

CASE FILE

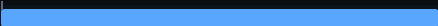
"Claude built her a trading bot"

An X post seen **7,996,185 impressions, 17,092 bookmarks**. Four sleeves: mean reversion on the S&P 500 and Nasdaq (15-minute bars), momentum on Bitcoin (1-hour bars), and trend following on gold, with ATR sizing and a 1% risk stop. Same rules, three different data setups:

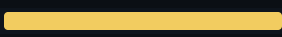
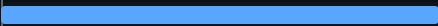
Daily bars, since 2019 2019-01-02 → 2026-10-16 · 230 trades

The bot		+370.7%
SPY buy & hold		+244.4%

Source timeframes: 15m S&P/Nasdaq, 1h BTC & gold ~60 days, from 2026-07-27 · 215 trades

The bot		-1.7%
SPY buy & hold		+3.5%

1-hour bars on every sleeve ~2 years, from 2024-10-28 · 793 trades

The bot		-22.0%
SPY buy & hold		+34.6%

Bars are scaled within each panel. Our engine, \$10,000 start, next-bar fills, 5 bps slippage on ETFs and 10 bps on crypto. Oil and the post's correlation filter are omitted in every version. Free intraday data only goes back about 60 days (15m) and about 2 years (1h).

Same rules, same dates, different bars

WINDOW	DAILY BARS	INTRADAY BARS
~60 days (15m / 1h)	+3.0%	-1.7%
~2 years (1h)	+26.2%	-22.0%

Daily-bar version of the bot run over the same windows as the intraday re-runs.

What happened?

On daily bars the bot looks brilliant: **+370.7%**, beating SPY's +244.4%. On the bars the post actually described, it lost money, while simply holding SPY made money over the same windows. On 1-hour bars it won only 18% of 793 trades.

Why claims fall apart

- **Vague rules** that can be coded many ways
- **Wrong timeframe** or cherry-picked dates
- **No fees or realistic fills**
- **No benchmark**: buy-and-hold quietly did better

The lesson

A backtest is one answer to one very specific question. Change the timeframe, the dates, the fees, or one fuzzy rule and you can get the opposite answer. That's why this kit is built around **checking**, not believing, including our own numbers.

The 5-step safe build workflow

This is the exact loop we use on every bot on the leaderboard. It's boring on purpose. Boring is how you avoid losing money to a bug.

Prompt → **Backtest** → **Look-ahead check** → **vs. buy-and-hold** → **Paper only**

1

Prompt

Ask for a **backtestable** script with every rule spelled out: entry, exit, sizing, data source, start date, fees, and "fill at the next bar's open". Vague prompts get vague bots. Use the tested prompts on pages 6–10 as templates.

✓ **Done when:** Prompt is specific enough that two people would code the same bot.

2

Run the backtest

Run the script yourself (free Python, pandas and yfinance are enough). Don't trust numbers the chatbot "reports" without running code. Make sure it runs end to end: one of our test scripts crashed on its last line.

✓ **Done when:** You have a trade log CSV and an equity curve on your own machine.

3

Check for look-ahead

Make sure the bot only uses information it would have had at the time. Signals on today's close, orders at tomorrow's open. Page 4 has a 60-second checklist.

✓ **Done when:** No future data leaks into any signal, indicator, or model fit.

4

Compare to buy-and-hold

Put the bot next to simply buying and holding the same asset over the same dates. Look at max drawdown too, not just total return.

✓ **Done when:** You can say in one sentence why the bot is worth the extra complexity.

5

Paper trade only

Run it forward on a paper (simulated money) account for weeks, not days. Keys in environment variables, paper endpoint hard-coded. Never point it at real money because a backtest looked good.

✓ **Done when:** Paper results roughly match what the backtest predicted.

Golden rule: each step can send you back to step 1. If the bot fails a check, fix the prompt and rerun. Don't "fix" it by tweaking rules until the backtest looks good. That's curve-fitting, and it's how great-looking backtests turn into bad paper results.

Two checks that kill most bots

Steps 3 and 4 are where most "I made 300% with a bot" stories end. Both take minutes.

Check 1 · Look-ahead in 60 seconds

Look-ahead bias means the backtest secretly uses information from the future, like buying at today's open based on today's close. It makes almost any strategy look amazing. Read the code (or paste it back into the chatbot) and look for:

- **Same-bar fills.** Signal computed on a bar's close, but the trade fills on that same bar. It should fill on the *next* bar's open.
- **Shifting the wrong way.** `shift(-1)` pulls tomorrow's value into today. Signals should use `shift(1)` or later.
- **Fitting on the whole history.** Scaling data, picking parameters, or training a model (like an HMM) on all the data, then "testing" on that same data. Fit only on the past, then step forward (walk-forward).
- **Centered or future-peeking indicators.** Rolling windows with `center=True`, or highs/lows that include the current bar when the rule says "prior".
- **Ask the bot to audit itself:** "List every line where a signal could use data not available at the time of the trade."

In our ChatGPT test, none of the 5 scripts had a look-ahead bug, because the prompts spell out "signals on the close, fill at the next day's open". That sentence is doing a lot of work. Keep it.

Check 2 · Did it beat just holding?

Our leaderboard backtests since 2019-01-02, each starting with \$10,000, against buy-and-hold of the bot's own market:

BOT	BOT RETURN	BOT MAX DD	MARKET	BUY & HOLD	B&H MAX DD
The 'Claude Built Her a Trading Bot' Multi-Asset Bot	+370.7%	-13.8%	SPY	+244.4%	-33.7%
The Reddit Dip-Ladder Crypto Bot	+266.2%	-44.0%	BTC-USD	+1,970.0%	-76.6%
The 14-Minute Claude Bot (20/50 MA Crossover)	+92.1%	-28.2%	SPY	+244.4%	-33.7%
The 34-Minute Regime Bot (HMM Bull/Bear/Sideways)	+83.8%	-11.0%	SPY	+244.4%	-33.7%
The 'Beat the Market' MA Mean-Reversion Bot	+83.0%	-47.9%	BTC-USD	+1,970.0%	-76.6%

1 of 5 bots beat buy-and-hold on total return, and that one is the daily-bar version of the bot that lost money on intraday bars (page 2). Several bots had much smaller drawdowns than holding, which can be a real reason to use one. Decide which matters to you *before* you look.

The 5 tested prompts

Each prompt rebuilds one viral bot from our leaderboard. We pasted the original versions into ChatGPT, ran the scripts it wrote *unmodified*, and compared the results with our own engine.

How we tested. ChatGPT Free (GPT-5.x "Instant" model; exact name not shown), 2026-10-16. Scripts run as-is in a fresh Python environment (pandas, yfinance, hmmlearn). No broker keys; paper mode off. Then we compared returns, drawdown and trade count with our engine and traced every gap to a cause.

PROMPT	GRADE	CHATGPT RETURN	OUR ENGINE	TRADES (GPT / OURS)
20/50 Moving-Average Crossover (SPY)	PARTIAL	+92.1%	+92.1%	16 / 16
HMM Regime-Switching Bot (SPY)	PARTIAL	+105.3%	+83.8%	67 / 64
Multi-Asset Mean-Reversion + Momentum + Trend Bot	PARTIAL	+320.9%	+370.7%	229 / 230
Crypto Dip-Ladder Bot (BTC + ETH)	PARTIAL	+141.4%	+266.2%	309 / 309
Moving-Average Mean-Reversion Bot (BTC)	PASS	+83.0%	+83.0%	44 / 44

What the grades mean

PASS ran cleanly and matched our engine.
PARTIAL usable, but had a bug, a gap we had to explain, or a "paper mode" that didn't really trade the strategy.

The honest fine print

After the test we **tightened every prompt's wording** to close the gaps we found. The copy-ready versions on the next pages are the tightened ones, and **they have not been retested yet**. Treat them as better starting points, not guarantees.

Which one first? Start with **Prompt 1** (MA crossover): the simplest rules, only 16 trades to check by hand, and ChatGPT matched our numbers exactly. Then try **Prompt 5**, the only Pass. Save the HMM bot (Prompt 2) for when you're comfortable debugging.

How to use them: copy one prompt into ChatGPT or Claude, run the script it gives you, then walk through steps 2–5 of the workflow. If your numbers differ from ours, that's not failure. Finding out *why* is the whole skill.

PROMPT 1/5

CHATGPT TEST: PARTIAL

20/50 Moving-Average Crossover (SPY)

Rebuilds **The 14-Minute Claude Bot (20/50 MA Crossover)** · market: SPY (S&P 500 ETF)

COPY-READY PROMPT · tightened wording, not yet retested

works in ChatGPT · Claude · Grok

You are a careful quant developer. Build me a backtestable trading bot for SPY using a 20-day / 50-day simple moving-average crossover: go fully long when the 20-day SMA closes above the 50-day SMA, and sell to cash when it closes below. If the two SMAs are exactly equal, keep the current position. Backtest from 2019-01-02 to today. Constraints: Python 3.11, pandas, yfinance for free daily data. Long-only, no leverage. Compute signals on the daily close and fill at the NEXT day's open (no look-ahead). Charge 5 bps slippage per fill on ETFs and 10 bps on crypto. Start with \$10,000. Output a trade log CSV (entry/exit date, price, qty, P&L, exit reason) and an equity curve, and compare against buy-and-hold of the same asset. Then add an optional Alpaca PAPER trading mode (endpoint <https://paper-api.alpaca.markets> only) that reads keys from environment variables, computes the same signal from the latest completed daily bar, and automatically places the matching order with the same sizing (and stops, if any) as the backtest; add a `--dry-run` flag that only prints the order. Never hard-code keys and never use a live trading endpoint. Make sure the script runs end to end without errors (all imports present) before you answer.

▲ THE ONE GOTCHA

Check that it runs all the way through.

The original script crashed on its very last line because it called `json.dumps` without `import json`. The backtest files were still written and the numbers matched ours exactly, so it's easy to miss. The tightened prompt asks for all imports and an end-to-end run, but run it yourself and read the last lines of output.

TEST RECEIPT · 2026-10-16

	CHATGPT	OUR ENGINE
Total return	+92.1%	+92.1%
Max drawdown	-28.2%	-28.2%
Trades	16	16

Original wording, backtest from 2019-01-02, \$10,000 start.
Look-ahead bug: none found.

CHECK YOUR RUN AGAINST THE LEADERBOARD

Our engine, 2019-01-02 → 2026-10-16: **+92.1%** return, -28.2% max drawdown, 16 trades. Buy-and-hold SPY: **+244.4%**. Your numbers will drift a little as new bars arrive; big gaps mean a rule was read differently.

VIRAL SOURCE

685,001 impressions, 1,660 bookmarks (X API, read 2026-10-16)

PROMPT 2/5

CHATGPT TEST: PARTIAL

HMM Regime-Switching Bot (SPY)

Rebuilds The 34-Minute Regime Bot (HMM Bull/Bear/Sideways) · market: SPY

COPY-READY PROMPT · tightened wording, not yet retested

works in ChatGPT · Claude · Grok

You are a careful quant developer. Build a regime-switching trading bot for SPY. Features: daily log return and 20-day rolling volatility, standardized using ONLY the training window. Fit a 3-state Gaussian HMM (hmmlearn, diagonal covariance) on the previous 756 trading days and refit every 63 days (walk-forward, never train on future data). Label states by the fitted model's mean of the (standardized) return feature (model.means_): lowest = bear, middle = sideways, highest = bull. Each day's regime = the last state of model.predict() on the trailing 250 days ending today (causal; standardized with the training-window mean/std). Use random_state=7, n_iter=200, min_covar=1e-3. Rules: bull = hold SPY; bear = cash; sideways = RSI(2) mean reversion (buy when RSI(2) < 10, sell when RSI(2) > 70). Backtest from 2019-01-02. Constraints: Python 3.11, pandas, yfinance for free daily data (use split- and dividend-adjusted OHLC, auto_adjust=True, for both signals and fills). Long-only, no leverage. Compute signals on the daily close and fill at the NEXT day's open (no look-ahead). Charge 5 bps slippage per fill on ETFs and 10 bps on crypto. Start with \$10,000. Output a trade log CSV (entry/exit date, price, qty, P&L, exit reason) and an equity curve, and compare against buy-and-hold of the same asset. Then add an optional Alpaca PAPER trading mode (endpoint <https://paper-api.alpaca.markets> only) that reads keys from environment variables, computes the same signal from the latest completed daily bar, and automatically places the matching order with the same sizing (and stops, if any) as the backtest; add a --dry-run flag that only prints the order. Never hard-code keys and never use a live trading endpoint. Make sure the script runs end to end without errors (all imports present) before you answer.

▲ THE ONE GOTCHA

Unpinned model details move the results.

HMM regime bots are sensitive to details a prompt can leave open: how states get labeled, the random seed, and how "today's" state is picked. ChatGPT's version was about 21 points above ours for that reason. The tightened prompt pins them all (labels from `model.means_`, `random_state=7`, a causal trailing window). If you change any of them, expect different numbers.

TEST RECEIPT · 2026-10-16

	CHATGPT	OUR ENGINE
Total return	+105.3%	+83.8%
Max drawdown	-10.3%	-11.0%
Trades	67	64

Original wording, backtest from 2019-01-02, \$10,000 start.
Look-ahead bug: none found.

CHECK YOUR RUN AGAINST THE LEADERBOARD

Our engine, 2019-01-02 → 2026-10-16: **+83.8%** return, -11.0% max drawdown, 64 trades. Buy-and-hold SPY: **+244.4%**. Your numbers will drift a little as new bars arrive; big gaps mean a rule was read differently.

VIRAL SOURCE

450,974 impressions, 1,332 bookmarks (X API, read 2026-10-16)

PROMPT 3/5

CHATGPT TEST: PARTIAL

Multi-Asset Mean-Reversion + Momentum + Trend Bot

Rebuilds The 'Claude Built Her a Trading Bot' Multi-Asset Bot · market: SPY, QQQ, BTC-USD, GLD

COPY-READY PROMPT · tightened wording, not yet retested

works in ChatGPT · Claude · Grok

You are a careful quant developer. Build a multi-asset daily-bar trading bot with four sleeves: (1) SPY and QQQ mean reversion: buy when the close is below the lower 20-day, 2-standard-deviation Bollinger band (sample standard deviation, pandas default ddof=1); exit when it closes at or above the 20-day mean. (2) BTC-USD momentum breakout: buy when the close exceeds the prior 20-day high; exit when it closes below the prior 10-day low. (3) GLD trend following: buy when close > 50-day SMA > 200-day SMA; exit when close < 50-day SMA. Position sizing: risk 1% of total equity per trade with a hard stop 1 ATR(14) below entry (cap stop at 20%), and no sleeve larger than 25% of equity. Check stops against each day's low. Compute every indicator on each asset's own full daily history (for BTC-USD that includes weekends, so its 20-day high, 10-day low and ATR use 7-day-a-week bars), then make trading decisions and fills only on US equity trading days. Keep any position still open at the end marked to market (don't force a final sale). Backtest from 2019-01-02. Constraints: Python 3.11, pandas, yfinance for free daily data. Long-only, no leverage. Compute signals on the daily close and fill at the NEXT day's open (no look-ahead). Charge 5 bps slippage per fill on ETFs and 10 bps on crypto. Start with \$10,000. Output a trade log CSV (entry/exit date, price, qty, P&L, exit reason) and an equity curve, and compare against buy-and-hold of the same asset. Then add an optional Alpaca PAPER trading mode (endpoint <https://paper-api.alpaca.markets> only) that reads keys from environment variables, computes the same signal from the latest completed daily bar, and automatically places the matching order with the same sizing (and stops, if any) as the backtest; add a --dry-run flag that only prints the order. Never hard-code keys and never use a live trading endpoint. Make sure the script runs end to end without errors (all imports present) before you answer.

▲ THE ONE GOTCHA

Bitcoin trades on weekends. Stocks don't.

ChatGPT computed Bitcoin's 20-day high, 10-day low and ATR on US trading days only, and used a slightly different Bollinger standard deviation. Rerunning our engine that same way gave +320.7%, so the whole gap was the wording. The tightened prompt says to compute BTC indicators on its full 7-day history but trade only on US sessions. Also remember page 2: this bot lost money on intraday bars.

TEST RECEIPT · 2026-10-16

	CHATGPT	OUR ENGINE
Total return	+320.9%	+370.7%
Max drawdown	-17.2%	-13.8%
Trades	229	230

Original wording, backtest from 2019-01-02, \$10,000 start.
Look-ahead bug: none found.

CHECK YOUR RUN AGAINST THE LEADERBOARD

Our engine, 2019-01-02 → 2026-10-16: **+370.7%** return, -13.8% max drawdown, 230 trades. Buy-and-hold SPY: **+244.4%**. Your numbers will drift a little as new bars arrive; big gaps mean a rule was read differently.

VIRAL SOURCE

7,996,185 impressions, 17,092 bookmarks (X API, read 2026-10-16)

PROMPT 4/5

CHATGPT TEST: PARTIAL

Crypto Dip-Ladder Bot (BTC + ETH)

Rebuilds The Reddit Dip-Ladder Crypto Bot · market: BTC-USD, ETH-USD

COPY-READY PROMPT · tightened wording, not yet retested

works in ChatGPT · Claude · Grok

You are a careful quant developer. Build a crypto dip-buying ladder bot for BTC-USD and ETH-USD on daily bars. Use 5 slots per coin (10 total). Size each new slot at the moment of entry as current total equity (cash + open slots marked at the last close) divided by 10, capped at available cash, so slot size grows and shrinks with the account. When a coin closes above its 100-day SMA AND is down 3% or more on the day, buy one empty slot for that coin at the next open. Each slot has a +5% profit-target sell (check against each day's high). No stop-loss. Report open (unrealized) slots separately from closed trades. Backtest from 2019-01-02. Constraints: Python 3.11, pandas, yfinance for free daily data. Long-only, no leverage. Compute signals on the daily close and fill at the NEXT day's open (no look-ahead). Charge 5 bps slippage per fill on ETFs and 10 bps on crypto. Start with \$10,000. Output a trade log CSV (entry/exit date, price, qty, P&L, exit reason) and an equity curve, and compare against BTC-USD buy-and-hold (and also show ETH-USD buy-and-hold). Then add an optional Alpaca PAPER trading mode (endpoint <https://paper-api.alpaca.markets> only) that reads keys from environment variables, computes the same signal from the latest completed daily bar, and automatically places the matching order with the same sizing (and stops, if any) as the backtest; add a --dry-run flag that only prints the order. Never hard-code keys and never use a live trading endpoint. Make sure the script runs end to end without errors (all imports present) before you answer.

▲ THE ONE GOTCHA

"Split equity into slots" can mean two things.

ChatGPT read it as fixed \$1,000 slots that never grow. We resize each slot to the current account. Our engine with fixed \$1,000 slots gives +141.4%, so the logic matched; the sizing rule alone was worth ~125 points. The tightened prompt defines slot size as current equity ÷ 10. Also watch the "100% win rate" (see page 11).

TEST RECEIPT · 2026-10-16

	CHATGPT	OUR ENGINE
Total return	+141.4%	+266.2%
Max drawdown	-36.6%	-44.0%
Trades	309	309

Original wording, backtest from 2019-01-02, \$10,000 start.
Look-ahead bug: none found.

CHECK YOUR RUN AGAINST THE LEADERBOARD

Our engine, 2019-01-02 → 2026-10-17: **+266.2%** return, -44.0% max drawdown, 309 trades. Buy-and-hold BTC-USD: **+1,970.0%**. Your numbers will drift a little as new bars arrive; big gaps mean a rule was read differently.

VIRAL SOURCE

Reddit comment in a top r/ai_trading thread
(engagement not measured)

PROMPT 5/5

CHATGPT TEST: PASS

Moving-Average Mean-Reversion Bot (BTC)

Rebuilds The 'Beat the Market' MA Mean-Reversion Bot · market: BTC-USD

COPY-READY PROMPT · tightened wording, not yet retested

works in ChatGPT · Claude · Grok

You are a careful quant developer. Build a mean-reversion bot for BTC-USD on daily bars: buy with all equity when the close is 7% or more below its 20-day SMA; sell when the close gets back to or above the 20-day SMA. Backtest from 2019-01-02. Constraints: Python 3.11, pandas, yfinance for free daily data. Long-only, no leverage. Compute signals on the daily close and fill at the NEXT day's open (no look-ahead). Charge 5 bps slippage per fill on ETFs and 10 bps on crypto. Start with \$10,000. Output a trade log CSV (entry/exit date, price, qty, P&L, exit reason) and an equity curve, and compare against buy-and-hold of the same asset. Then add an optional Alpaca PAPER trading mode (endpoint <https://paper-api.alpaca.markets> only) that reads keys from environment variables, computes the same signal from the latest completed daily bar, and automatically places the matching order with the same sizing (and stops, if any) as the backtest; add a `--dry-run` flag that only prints the order. Never hard-code keys and never use a live trading endpoint. Make sure the script runs end to end without errors (all imports present) before you answer.

▲ THE ONE GOTCHA

"Paper mode" that never places an order.

The backtest ran cleanly and matched ours exactly. But the paper-trading mode only checked the Alpaca paper account and never placed a trade. A bot that "connects" isn't the same as a bot that trades its strategy. The tightened prompt asks it to place the strategy's own orders, with a `--dry-run` flag to preview them first.

TEST RECEIPT · 2026-10-16

	CHATGPT	OUR ENGINE
Total return	+83.0%	+83.0%
Max drawdown	-47.9%	-47.9%
Trades	44	44

Original wording, backtest from 2019-01-02, \$10,000 start.
Look-ahead bug: none found.

CHECK YOUR RUN AGAINST THE LEADERBOARD

Our engine, 2019-01-02 → 2026-10-17: **+83.0%** return, -47.9% max drawdown, 44 trades. Buy-and-hold BTC-USD: **+1,970.0%**. Your numbers will drift a little as new bars arrive; big gaps mean a rule was read differently.

VIRAL SOURCE

4,995 impressions, 60 bookmarks (X API, read 2026-10-16)

The bot-claim BS-detector

Run any viral "my AI bot made X%" post through this list. Each box you tick is a reason to wait for receipts. Three or more? Scroll on.

☐ **Only a screenshot, no rules.** If you can't rebuild it from the post, you can't check it. Ask for exact entry, exit and sizing rules.

☐ **No buy-and-hold comparison.** "+90%" sounds great until you learn holding SPY did +244.4% over the same dates (page 4).

☐ **Suspiciously short or hand-picked dates.** A few great months prove nothing. Ask for results back to at least 2019, through crashes and chop.

☐ **No fees, slippage or fill rules.** Filling at the same close that triggered the signal usually hides look-ahead. Real fills come later, at worse prices.

☐ **Win rate is the headline.** Our dip-ladder backtest won **100%** of closed trades and still had a **-44.0%** max drawdown. With no stop-loss, losers just stay open: it ended holding 5 underwater ETH slots, the oldest down -45.6%.

☐ **Timeframe mismatch.** Results on daily bars say nothing about the 15-minute version. The same multi-asset rules went from +370.7% to -22.0% (page 2).

☐ **Max drawdown not shown.** Ask "what was the worst peak-to-trough drop?" Could you have sat through it without pulling the plug?

☐ **"The AI found it" with no code.** A chatbot can describe results it never computed. Numbers only count if code produced them and you can rerun it.

☐ **Live money from day one.** Anyone skipping the paper-trading stage is either selling something or about to learn an expensive lesson.

☐ **There's a course, signal group or "link in bio".** Not automatically a scam, but now the claim is also an ad. Raise the bar for proof.

Copy-paste reply for any bot post:

"Cool result! What are the exact entry, exit and sizing rules? What dates, and what fees and slippage? How did buy-and-hold do over the same period, and what was the max drawdown? Is this a backtest, paper trading, or live money?"

Real builders answer these easily. Sellers change the subject.

Score it: 0–1 ticks: worth rebuilding and testing yourself. 2–3: interesting idea, unproven claim. 4+: marketing, not a trading result.

How to paper trade safely

Paper trading runs your bot against live market prices with simulated money. It's the only honest way to see whether a backtest survives contact with the real world, and it costs nothing.

1 Use Alpaca's paper endpoint, and only that

Every prompt in this kit pins the bot to `https://paper-api.alpaca.markets`. Alpaca's paper trading: **Yes, default; \$100k simulated**; minimum balance **\$0** (per Alpaca's docs, checked 2026-10-16). Hard-code the paper URL so the script *can't* reach a live account by accident.

2 Keys go in environment variables, never in code

Use paper-account keys only, and never paste keys into a chat, a screenshot or a shared file.

```
export ALPACA_API_KEY="your-paper-key"
export ALPACA_SECRET_KEY="your-paper-secret"
python3 bot.py --dry-run      # prints the orders, sends nothing
python3 bot.py               # sends to the PAPER account
```

3 Dry-run first, every time you change something

Read the printed orders. Is the size what you expected? Is there a stop? Our tests found "paper modes" that dropped the stops, used a fixed size, only bought (never sold), or never traded at all.

4 Same rules as the backtest, or it's a different bot

Paper mode should compute the signal from the latest *completed* bar and place the same order the backtest would, with the same sizing and stops.

5 Give it time, and keep a log

Days tell you nothing. Run for weeks and compare paper trades with what the backtest says should have happened over the same days. One paper account per bot keeps the P&L clean.

PRE-FLIGHT CHECKLIST · before your first paper order

- | | | |
|---|--|--|
| ☐ Backtest runs end to end | ☐ Look-ahead check passed | ☐ Compared to buy-and-hold |
| ☐ Keys are paper keys, in env vars | ☐ URL is paper-api.alpaca.markets | ☐ Dry-run orders look right |

Never live. This kit is about learning and testing, not trading real money. Don't connect an AI agent or bot to a funded account because a backtest or a few paper weeks looked good. Some broker AI connections place trades with no approval step by default; check our Broker AI Tracker before you connect anything.

Free & cheap tools to start with

You can build and test every bot in this kit for \$0: free Python, pandas, yfinance for daily data, a chatbot's free tier, and an Alpaca paper account. When you want more, these are reasonable next steps.

\$0

Python + pandas + yfinance (free, open source)

\$0

A chatbot's free tier (our test used ChatGPT Free)

\$0

Alpaca paper account: \$100k simulated, \$0 minimum

TOOL	CATEGORY	WHY IT'S USEFUL	PRICE (AS LISTED)	PRICE CHECK
TradingView tradingview.com	Research & Analysis	Charts to eyeball your bot's signals and sanity-check trades.	Free; paid from \$12.95/mo (Essential, billed annually)	Verified checked 2026-10-16
QuantConnect quantconnect.com	Multi-Tool Platform	Browser-based backtesting platform when you outgrow a single script.	Free plan; paid tiers are configurable (no single list price shown)	Verified checked 2026-10-16
Superalgos superalgos.org	AI Trading Bot	Open-source bot platform you run yourself.	Free, open source	Verified checked 2026-10-16
Composer composer.trade	AI Trading Bot	No-code strategy builder, if you'd rather not touch Python.	Free (Starter); paid from \$10/mo (Advanced)	Verified checked 2026-10-16
Finviz finviz.com	Stock Screener	Stock screener for finding what to test next.	Free; Elite \$24.96/mo billed annually (\$39.50 monthly)	Verified checked 2026-10-16
Koyfin koyfin.com	Research & Analysis	Research dashboards and fundamentals.	Free; paid from \$39/mo	Verified checked 2026-10-16
TradesViz tradesviz.com	Portfolio Manager	Trading journal to log and review paper trades.	Free; paid from \$14.99/mo billed annually (\$19.99 monthly)	Verified checked 2026-10-16
Perplexity Finance perplexity.ai/finance	Research & Analysis	AI research assistant for quick market questions.	Free; Pro \$17/mo billed annually (\$20 monthly)	Verified (search snapshot) checked 2026-10-16

Prices change. Prices checked 2026-10-16 against each company's own pricing page (linked per row). "Search snapshot" means read from the official page via search because the page blocked direct fetching. Check the vendor's page before you buy.

Not endorsements. We haven't tested these tools as bots yet. They're listed so you know where to look, and none of these are affiliate links. Full list of 54 tools: getbotreceipts.com/tools

WHAT'S NEXT

See the receipts, not the screenshots.

Every viral bot we test goes on a public leaderboard: rules, data, every trade, and how it did against simply holding. Backtests since 2019-01-02, plus a forward paper run that started 2026-10-16.

THE LEADERBOARDgetbotreceipts.com

All 5 bots, trade logs, equity curves and the multi-asset intraday re-run.

THE PROMPTSgetbotreceipts.com/prompts

Copy the latest versions and check their test status.

THE WEEKLY RECEIPTS · FREEgetbotreceipts.com/newsletter

One email a week: what the viral AI trading bots actually did, new bots added, and broker AI-access changes.

BROKER AI TRACKERgetbotreceipts.com/brokers

Which brokers let AI place orders, which have paper modes, and which ask for approval.

YOUR NEXT 30 MINUTES

1. Copy Prompt 1 (page 6) into ChatGPT or Claude.
2. Run the script and open the trade log.
3. Compare your numbers with the leaderboard and buy-and-hold.
4. Subscribe to the weekly receipts so you hear when we retest the prompts.

Bot Receipts paper-trades and backtests publicly described strategies. Backtests are hypothetical and simulated; past results, real or simulated, don't predict future results. Nothing here is a recommendation to buy or sell anything. © 2026 Bot Receipts.