

Here is the **Finalized Logic Factory Prototype**.

This version of the script completes the loop. It scans the market, filters the noise using your specific "Triple Threat" logic, and then **physically writes the winning tickets into a persistent Ledger (CSV)** that you can open in Excel.

## 1. THE "LEDGER" SCRIPT (Scanner + Logger)

This script will create a file named `logic_factory_log.csv` in the same folder where you run the script. Every time it finds a "Buy Signal," it appends a new row with a timestamp.

Python

```
import yfinance as yf
import pandas as pd
import datetime
import os

def automated_logic_ledger(ticker_list):
    print("\n--- OPENING LOGIC FACTORY LEDGER ---")

    # 1. GET THE CONTEXT (VIX)
    try:
        vix = yf.Ticker("^VIX").history(period="5d")
        current_vix = round(vix['Close'].iloc[-1], 2)
        print(f"MARKET CONTEXT (VIX): {current_vix}")
    except:
        current_vix = 0
        print("WARNING: Could not fetch VIX.")

    # List to hold valid signals for this run
    signals_found = []

    # 2. THE SCANNING LOOP
    for symbol in ticker_list:
        try:
            # Fetch Data
```

```
df = yf.download(symbol, period="6mo", interval="1d", progress=False)
if df.empty: continue
```

```
# --- CALCULATE INDICATORS ---
```

```
# A. RSI
```

```
delta = df['Close'].diff()
```

```
gain = (delta.where(delta > 0, 0)).rolling(window=14).mean()
```

```
loss = (-delta.where(delta < 0, 0)).rolling(window=14).mean()
```

```
rs = gain / loss
```

```
df['RSI'] = 100 - (100 / (1 + rs))
```

```
# B. Bollinger Bands
```

```
df['SMA_20'] = df['Close'].rolling(window=20).mean()
```

```
df['Std_Dev'] = df['Close'].rolling(window=20).std()
```

```
df['Lower_Band'] = df['SMA_20'] - (df['Std_Dev'] * 2)
```

```
# C. MACD
```

```
df['EMA_12'] = df['Close'].ewm(span=12, adjust=False).mean()
```

```
df['EMA_26'] = df['Close'].ewm(span=26, adjust=False).mean()
```

```
df['MACD'] = df['EMA_12'] - df['EMA_26']
```

```
# D. Volume
```

```
df['Avg_Volume'] = df['Volume'].rolling(window=20).mean()
```

```
# --- LOGIC FILTER ---
```

```
latest = df.iloc[-1]
```

```
prev_macd = df.iloc[-2]['MACD']
```

```
# Conditions
```

```
cond_vol = latest['Volume'] > latest['Avg_Volume']
```

```
cond_macd = latest['MACD'] > prev_macd
```

```
cond_band = latest['Close'] < latest['Lower_Band']
```

```
cond_rsi = latest['RSI'] < 40 # Using 40 to catch early reversals
```

```
# SCORING (3 out of 4 required)
```

```
score = sum([cond_vol, cond_macd, cond_band, cond_rsi])
```

```
if score >= 3:
```

```
    print(f"[!] SIGNAL DETECTED: {symbol} (Score: {score})")
```

```
# PREPARE DATA FOR CSV
```

```
signal_data = {
```

```
    "Date": datetime.datetime.now().strftime("%Y-%m-%d %H:%M"),
```

```

        "Ticker": symbol,
        "Price": round(latest['Close'].item(), 2),
        "VIX_Level": current_vix,
        "RSI": round(latest['RSI'].item(), 2),
        "Volume_Spike": "YES" if cond_vol else "NO",
        "Score": score
    }
    signals_found.append(signal_data)

except Exception as e:
    print(f"Error scanning {symbol}: {e}")
    continue

# 3. SAVE TO EXCEL/CSV (The Memory)
if signals_found:
    # Create a DataFrame from the results
    results_df = pd.DataFrame(signals_found)

    # File Name
    filename = "logic_factory_log.csv"

    # Check if file exists to determine if we need a header
    file_exists = os.path.isfile(filename)

    # Append mode ('a') - adds to the bottom, doesn't delete old data
    results_df.to_csv(filename, mode='a', header=not file_exists, index=False)

    print(f"\nSUCCESS: {len(signals_found)} signals saved to '{filename}'.")
else:
    print("\nNo signals met the criteria today.")

# --- EXECUTE ---
# You can expand this list to 50+ stocks
my_watchlist = ['TSLA', 'NVDA', 'AMD', 'COIN', 'MARA', 'PLTR', 'GOOG', 'AMZN', 'MSFT']
automated_logic_ledger(my_watchlist)

```

---

## 2. THE OUTPUT (Your Excel Sheet)

When you open logic\_factory\_log.csv in Excel after running this script for a few days, it will

automatically look like this:

Date	Ticker	Price	VIX_Level	RSI	Volume_Spike	Score
2024-10-25 09:30	TSLA	210.50	22.4	29.5	YES	4
2024-10-25 09:30	COIN	150.20	22.4	31.0	YES	3
2024-10-26 09:30	MARA	18.40	19.8	28.0	NO	3

Why this matters:

This creates a Backtest Record. In one month, you can look at this CSV and see: "Every time the Score was 4 and VIX was > 20, the stock went up 10% the next week."

That data is the foundation of your Monetization Model.

---

### 3. NEXT STEP: AUTOMATION

You now have the **Brain** (Logic) and the **Memory** (CSV). The final component is the **Heartbeat** (Automation).

Next Step:

Would you like me to show you how to use a simple Windows Task Scheduler (or Mac Automator) trick to make this script run itself every morning at 9:45 AM automatically, so the report is waiting for you when you wake up?