1. Scenario Builder – Equity Valuation Tool

* Developed an interactive Streamlit application to test investment theses by adjusting core valuation drivers (growth, margins, WACC, and multiples).
* Combines discounted cash flow (DCF) modelling with relative valuation techniques to generate base, upside, and downside cases.
* Includes sensitivity analysis features to identify which assumptions have the strongest impact on valuation outcomes.
* Produces clear visual outputs — valuation ranges, scenario comparisons, and charts of driver impacts — to support discretionary investment decisions.
* Example application: applied to Nike and Ralph Lauren to illustrate thesis-driven scenario outcomes.

2. Trading Analysis & Backtesting Tool

* Built a Python-based framework to evaluate discretionary trading strategies across equities, FX, and commodities.
* Implements backtesting modules for momentum, volatility filters, and event-driven strategies, using pandas/numpy/matplotlib.
* Integrates performance and risk metrics such as Sharpe ratio, volatility, and maximum drawdown to measure robustness.
* Outputs visual dashboards including cumulative return charts, drawdown plots, and trade-level distributions.
* Designed as a decision-support tool for discretionary investors to stress-test strategies under different market conditions.

3. Macro/Commodity Market Study

* Conducted data-driven studies of real macroeconomic and commodity market events to link fundamentals with trading implications.
* Example projects:
  + Analysis of the proposed 39% US tariff on Swiss gold exports, comparing spot vs. futures dynamics and implications for flows.
  + Study of copper arbitrage opportunities between US and LME markets, highlighting cross-exchange pricing premiums.
* Deliverables include Jupyter notebooks with cleaned datasets, graphs of price spreads, and concise commentary on the trading takeaways.
* Provides evidence of the ability to combine data analysis, policy developments, and trading opportunities in real markets.

4. Risk & Portfolio Metrics Library

* Created a Python library of reusable risk and portfolio analysis functions commonly used by traders and portfolio managers.
* Implements volatility calculations, Sharpe ratio, maximum drawdown, and Value-at-Risk (historical and parametric).
* Includes example notebooks applying the functions to S&P 500 data and FX pairs, showing real use cases.
* Designed for modularity so functions can be plugged into backtesting engines, dashboards, or portfolio management workflows.
* Accompanied by a concise README explaining each metric, its interpretation, and when traders use it in practice.

**Evaluation:**

Equity valuation:

* DCF must be adjusted, made fully interactive / stock dependent
* Calibration file needs to be checked and implemented in dcf file
* Need to create visual dashboard of analysis

Maybe use yfinance data for people to be able to check it / have it “open source”

**Project description:**

Data:

* Open source

3 analyses: DCF, Analyst estimates, Technical analysis

DCF:

* Industry calibration (tech, banking, industrial, automobile etc.)
  + Wacc, growth, multiples
  + Peer comparison (for multiples, growth, etc.)
* Careful for very large and very small stocks: DCF might not be accurate at all

Analyst estimates:

* Gather info from yfinance, investing.com, directly from banks analyses?, any open source / free source

Technicals:

* MAs, Bollinger bands, any relevant technical analysis method (can choose from a side pane which to show)
* Use them to identify trends and suggest where the stock may be going

Recommendation: combination of all 3 analyses

* Buy / Hold / Sell recommendation based on combination
* Expected target price at defined horizon based on combination

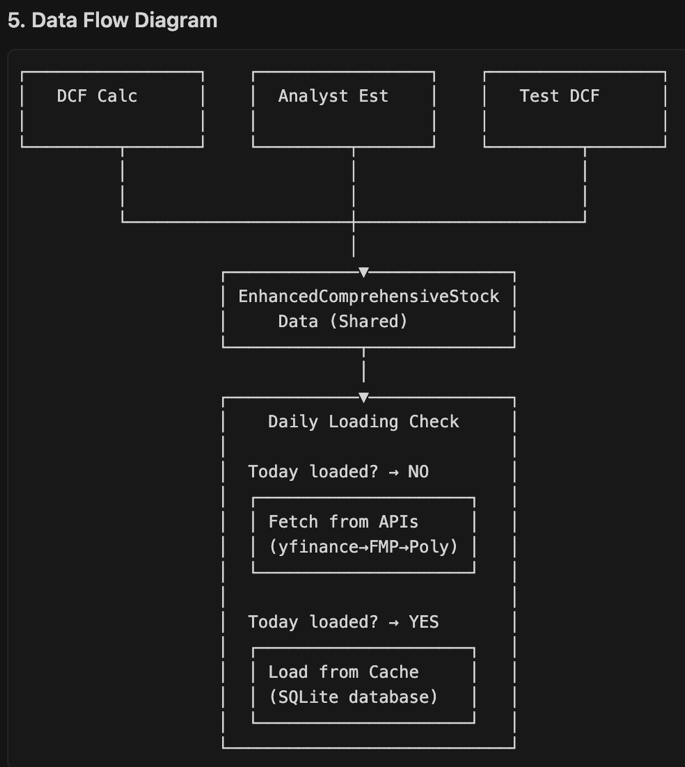
**Files breakdown:**

* DCF calibration
* DCF calculation
* Analyst estimates data + formatting
* Analyst estimates recommendation
* Technical analysis
* Analyses combination
* Analyses recommendation
* Visual interface
* Pre-downloaded data to not overuse api calls (that can be updated daily automatically)

Next steps:

* DCF: needs brainstorming to arrive to ±20% of market range
* Analyst estimates sources brainstorming

Data:



Technical analysis

Phase 1: Core Essential Indicators (4 indicators)

1. Enhanced Moving Averages System

* SMA 20, 50, 200 (already implemented, enhance with crossovers)
* EMA 12, 26 (add for faster signals)
* Golden Cross/Death Cross detection
* Price position relative to MAs

2. MACD Enhancement

* MACD line, Signal line, Histogram (already implemented, enhance)
* MACD crossovers and divergences
* Zero line crossovers
* Signal strength scoring

3. RSI Enhancement

* RSI calculation (already implemented, enhance)
* Overbought (>70) and Oversold (<30) levels
* RSI divergences
* Trend analysis

4. Bollinger Bands Enhancement

* Upper, Middle, Lower bands (already implemented, enhance)
* Band squeeze and expansion detection
* Price position within bands
* Volatility analysis

Phase 2: Intermediate Indicators (3 indicators)

5. Stochastic Oscillator

* %K and %D lines
* Overbought/oversold conditions
* Stochastic crossovers
* Momentum confirmation

6. Average True Range (ATR)

* Volatility measurement
* Stop-loss calculations
* Risk assessment
* Position sizing

7. Support/Resistance Detection

* Automatic level identification
* Historical significance scoring
* Dynamic level updates
* Breakout detection

Phase 3 & 4: Advanced Analysis (3 indicators)

8. Candlestick Pattern Recognition

* 5-6 key patterns: Doji, Hammer, Shooting Star, Engulfing (Bullish/Bearish), Morning Star
* Pattern reliability scoring
* Confirmation requirements
* Signal integration

9. Multi-Indicator Signal System

* Buy/Sell/Hold signal generation
* Signal strength scoring (1-10)
* Confirmation across multiple indicators
* Risk level assessment

10. Risk Management & Performance

* Stop-loss calculations (ATR-based)
* Position sizing recommendations
* Risk-reward ratios
* Signal backtesting framework