ChatGPT

Market Sentiment Trading Model

1. What We Are Building

A **rule-based, automated trading system** that uses two market-wide sentiment gauges—**CBOE VIX** and **CNN Fear & Greed Index**—to issue daily BUY/SELL signals on a benchmark ETF (initially **SPY**). The project is meant to be small, transparent, and fully reproducible so it can serve as a portfolio piece and the foundation for more advanced strategies.

2. Why This Approach?

- **Behavioural edge** Extreme fear/greed levels have a well-documented tendency to mean-revert, offering statistical edge without complex feature engineering.
- **Data availability** Both indicators are freely accessible via public endpoints/web-scrapes, ideal for a first project.
- **Explainability** Simple rules make it easy to discuss logic and justify decisions in interviews or reports.

3. Project Goals

Goal	Success Criterion
Executable demo	End-to-end pipeline that fetches data, generates signals, back-tests, and plots results
🎓 **Learning showcase**	Clean code, clear docs, and visuals suitable for GitHub/portfolio
📉 **Risk discipline**	Keep historical max drawdown within the 20-30 % tolerance band

4. Strategy v1.0 (MVP)

```
BUY  when VIX > 30  AND  CNN < 20  →  100 % SPY
SELL when VIX < 18  AND  CNN > 70  →  100 % CASH
otherwise HOLD
```

- **Position sizing**: full-switch ("all-in / all-out") for MVP.
- **Leverage / Shorting**: none at this stage.
- **Stop-loss**: deferred—first observe realised max drawdown.

5. Road-map & Milestones

Day	Module	Key Outputs
Day 1	Strategy spec	this README, Mermaid flowchart

Day	Module	Key Outputs
Day 2	`data_fetcher.py`	Raw VIX, CNN, SPY CSVs in `data/raw/`
Day 3	`signal_generator.py`	Daily signal column (BUY/SELL/HOLD)
Day 4	`backtester.py`	Equity curve, KPIs (CAGR, Sharpe, MDD)
Day 5	`visualizer.py` & Report	Annotated charts + PDF/Markdown summary

6. Repository Layout

```
market_sentiment_model/
├── data/
│   └── raw/            # downloaded CSV files
├── src/
│   ├── data_fetcher.py
│   ├── signal_generator.py
│   ├── backtester.py
│   └── visualizer.py
├── notebooks/          # exploratory work
├── README.md           # ← you are here
└── requirements.txt    # project dependencies
```

7. Future Enhancements (v1.x+)

- **Pyramiding / phased entries** to smooth drawdowns
- **Dynamic stop-loss & trailing exits**
- **Multi-asset universe** (e.g. QQQ, IWM, sector ETFs)
- **Docker / CI pipeline** for scheduled daily runs
- **Dashboard** (Streamlit/Gradio) for live monitoring

> Purpose summary: deliver a fully-documented, sentiment-driven trading model that is easy to reason about, quick to back-test, and impressive enough to show during internship or job interviews.