# Mini-Quant Desk Trading Bot Plan

## Repo Layout (Final)

repo/  
├─ apps/  
│ ├─ ingestor/ # live data → bus  
│ ├─ strategist/ # loads strategies, emits signals  
│ ├─ executor/ # ONLY process that talks to broker  
│ ├─ backtester/ # offline; same Strategy API  
│ └─ metrics/ # exporters, log shippers  
├─ core/  
│ ├─ strategy\_base.py # Strategy/Signal interfaces  
│ ├─ models.py # DTOs  
│ ├─ risk.py # sizing, OCO builder, kill switches  
│ ├─ bus.py # bus wrappers  
│ ├─ logging.py # structured JSON logger  
│ ├─ time.py # UTC helpers  
│ └─ settings.py # config loader  
├─ adapters/  
│ ├─ data/  
│ ├─ broker/  
│ └─ storage/  
├─ strategies/  
├─ services/  
├─ ml/  
├─ storage/  
├─ configs/  
├─ compose/  
├─ docker/  
├─ scripts/  
├─ tests/  
├─ Makefile  
├─ requirements.txt  
├─ .env.example  
└─ README.md

## Canonical Message Channels

- bars.{symbol} → ingestor → strategist  
- signals.new → strategist → arbiter/executor  
- orders.new → arbiter → executor  
- orders.acks → executor → DB/logs  
- fills.new → executor → DB/logs  
- metrics.\* → everyone → Prometheus/Loki

## Config Example

env: dev  
pdt\_mode: true  
universe: ["SPY","QQQ","AAPL","NVDA"]  
strategies:  
 - "strategies.orb:ORBStrategy"  
 - "strategies.vwap\_revert:VWAPRevert"  
 - "strategies.abcd:ABCDStrategy"

## Database Schema

Tables:  
- signals: id, ts, strategy, symbol, side, entry, stop, target, meta  
- orders\_outbox: client\_id, ts, payload, status  
- orders: id, client\_id, ts, symbol, side, qty, type, status, broker\_order\_id, meta  
- fills: id, ts, client\_id, price, qty, commission  
- strategy\_metrics: strategy, window\_start, window\_end, trades, wins, losses, avg\_r, sharpe, max\_dd

## Core Python Skeletons

Strategy base, Signal class, risk helpers, bus wrapper, settings loader.

## Strategies

Each strategy in its own file (abcd.py, orb.py, vwap\_revert.py). Plug-in system loads them dynamically.

## Services

Arbiter picks best signal; Scoring calculates Sharpe, win%, DD.

## Apps

Thin runners for ingestor, strategist, executor, backtester, metrics.

## Makefile

Quick commands: up, logs, down, test, lint.

## Startup Checklist

1) Create repo, paste structure  
2) Fill .env  
3) docker compose up  
4) Apply schema.sql  
5) Run backtester  
6) Enable ingestor/strategist/executor in paper mode  
7) Watch Grafana  
8) Add more strategies  
9) Deploy executor near exchange later

## Machine Learning Integration

- Features: ml/features.py  
- Models: ml/models.py  
- Training: ml/train.py  
- Saved models: ml/registry/  
- Strategies can call ML for scoring or run pure ML strategies  
- Executor always enforces risk & PDT rules