

## Quant X — Read-Only Dashboard

### Installation & Usage Guide (2025 Edition)

#### Version: Read-Only Monitoring Mode

Purpose: Display trade logs, equity curves, performance metrics

No strategy execution inside Streamlit

---

#### Overview — What This Dashboard Does

This dashboard is designed to:

- ✓ Auto-discover strategies from:

strategies\_runner/logs/<STRATEGY\_NAME>/

- ✓ Load each strategy's trade logs from:

trade\_log.csv

heartbeat.json (optional)

status.log (optional)

- ✓ Compute advanced performance metrics using  
utils/performance\_metrics.py

- ✓ Display:

- Trade tables
- Equity curve
- Drawdown curve
- Summary metrics
- Per-strategy analysis

#### Does NOT execute strategies.

The new design is **fully read-only** and simply reads files written by external strategy runners (ibapi or ib\_insync templates).

---

Link to Download the Read-Only Dashboard Code shall be provided.

---

 **1.**  Installation Quick Start

Follow these steps to set up the Quant X Read-Only Dashboard on your computer.

---

**1** Unzip the Download

After downloading:

1. Locate Quant\_X\_Dashboard\_Monitor.zip
2. Right-click → Extract All...
3. You will get a folder:

Quant\_X\_Dashboard\_Monitor/

---

**2** Open in VS Code

1. Open Visual Studio Code
2. Go to:

File → Open Folder...

3. Select the folder:

Quant\_X\_Dashboard\_Monitor

## 2. Folder Structure (Required)

Your project folder must look like this:

```
Quant_X_Dashboard_ReadOnly/
|
├── dashboard_read_only.py
├── requirements.txt
├── .env
|
├── utils/
│   ├── performance_metrics.py
│   └── client_id_manager.py
|
├── strategies_runner/
│   └── logs/
│       ├── SPY_0DTE/
│       │   ├── trade_log.csv
│       │   └── heartbeat.json (optional)
│       └── status.log (optional)
|
│   ├── IBM_Fourier/
│   └── IYW_GLD_org/
│       └── IYW_GLD_vanguard/
│           └── (More strategies...)
|
└── asset_test_runner.py (optional — for testing)
```

The **ONLY** file required per strategy is:

trade\_log.csv

The dashboard will appear empty if this folder structure is incorrect.

---

### 3. Install Python 3.12 (Required)

Streamlit + PyArrow compatibility works best with **Python 3.12**.

Download: <https://www.python.org/downloads/>

During installation (Windows):

- ✓ Add Python to PATH
  - ✓ Install py launcher
- 

### 4. Create a Virtual Environment (one-time)

Open VS Code → Open Folder → Quant\_X\_Dashboard\_ReadOnly

Open Terminal:

#### Windows:

```
python -m venv .venv  
.venv\Scripts\activate
```

#### macOS / Linux:

```
python3 -m venv .venv  
source .venv/bin/activate
```

You should see:

(.venv)

---

### 5. Select the .venv Interpreter in VS Code

Press:

Ctrl + Shift + P

Choose:

Python: Select Interpreter → .venv

---

## 6. Install Requirements

Inside the venv:

```
pip install --upgrade pip  
pip install -r requirements.txt
```

This installs:

- streamlit
  - pandas / numpy
  - altair
  - utils support
  - **No IB dependencies needed (read-only dashboard)**
- 

## 7. Configure .env (minimal)

Your read-only dashboard needs only:

```
# READ-ONLY DASHBOARD SETTINGS  
APP_ENV=local  
DASHBOARD_TIMEZONE=US/Eastern  
  
# Log root where strategies write CSVs  
LOG_ROOT=strategies_runner/logs
```

That's it.

**!** No IB\_HOST, IB\_PORT, IB\_CLIENT\_ID required.  
These are for strategy-runner templates only.

---

## 8. Run the Read-Only Dashboard

From VS Code terminal:

```
streamlit run dashboard_read_only.py
```

Browser opens:

<http://localhost:8501>

You'll see:

- Total PnL
  - Summary metrics
  - Per-strategy files
  - Logs from SPY\_0DTE, IBM\_Fourier, etc.
- 

## 9. Testing the Dashboard (Optional)

Use your test runner:

```
asset_test_runner.py
```

Run it:

```
python asset_test_runner.py
```

It writes:

```
strategies_runner/logs/ASSET/trade_log.csv
```

Refresh the dashboard → **ASSET** appears automatically.

---

## 10. What the Dashboard Computes

For each strategy:

- Closed P/L
- Open P/L
- Total P/L
- Number of Trades
- Win Rate
- Profit Factor

- CAGR
- Sharpe
- Sortino
- Volatility
- Max Drawdown
- Equity Curve
- Drawdown Curve

These are all computed from **trade\_log.csv**.

---

## 11. Requirements for Logs to Show Metrics Correctly

Trade logs **must** follow this format:

timestamp (UTC ISO format)

symbol

action (BUY/SELL)

price

quantity

pnl

duration

position

status (Filled)

ib\_order\_id

At minimum:

**✓ BUY row**

**✓ SELL row**

with pnl from SELL row.

**✓ 30+ trades**

**✓ at least several days of timestamps**

Otherwise:

- CAGR = 0
- Max DD = 0
- Sharpe = 0
- Volatility = 0

This is normal for small logs.

---

## 12. Adding New Strategies

To add a new strategy:

1. Create folder:

strategies\_runner/logs/<STRATEGY\_NAME>/

2. Ensure inside it:

trade\_log.csv

3. Dashboard automatically detects it — no configuration needed.
- 

## 13. Troubleshooting

### Strategy not showing

Check folder:

strategies\_runner/logs/<STRATEGY\_NAME>/trade\_log.csv

### Performance metrics = 0

Trade log too small (CAGR needs multiple days and >20 trades)

### Invalid comparison between datetime & timestamp

Make sure dashboard applies this fix:

```
df_ts["timestamp"] = pd.to_datetime(df_ts["timestamp"]).dt.tz_localize(None)  
today_utc = pd.Timestamp.utcnow().normalize().replace(tzinfo=None)
```

## Streamlit not found

Activate venv first:

```
.\venv\Scripts\activate
```

---

## 14. You Are Now Ready to Use the Dashboard

Every time you want to use it:

Activate venv

Run streamlit

View metrics

Windows:

```
.\venv\Scripts\activate  
streamlit run dashboard_read_only.py
```

macOS / Linux:

```
source .venv/bin/activate  
streamlit run dashboard_read_only.py
```