

ExecuTrade (Mock) – Binance Futures Order Bot

Candidate: *Rutuja Wagh*

1. Overview

ExecuTrade is a command-line Binance USDT-M Futures trading bot designed to simulate order placement using a mock structure.

Due to API restrictions and personal data privacy (KYC) concerns, this version uses mock functions that log simulated orders instead of executing real ones via the Binance API.

The system supports market, limit, OCO, and TWAP orders.

2. Architecture

- `main.py` – CLI interface to parse user input
- `market_orders.py` – Simulated market order logic
- `limit_orders.py` – Simulated limit order logic
- `advanced/oco.py` – Mock implementation of OCO
- `advanced/twap.py` – Mock implementation of TWAP
- `bot.log` – Structured logging of all order activities in JSON

No API keys or internet access required for testing.

3. Sample Commands

bash

CopyEdit

```
python src/main.py market BTCUSDT BUY 0.01
```

```
python src/main.py limit BTCUSDT SELL 0.01 30000
```

```
python src/main.py oco BTCUSDT SELL 0.01 31000 29500 29480
```

```
python src/main.py twap BTCUSDT BUY 0.04 4 3
```

4. Advanced Orders

- **OCO:** Places two linked mock orders (take-profit + stop-loss). Cancels one if the other is executed.

- **TWAP:** Splits a large trade into time-separated smaller trades using a loop with `sleep()`.
-

5. Logging Format

Each simulated order is appended as a JSON entry in `bot.log`.

Example fields:

```
{"order_id": "f1798a7d-936c-4009-83db-0fad198e8ad7", "timestamp": 1752247558132, "symbol": "BTCUSDT", "side": "BUY", "type": "MARKET", "quantity": 0.01, "status": "FILLED", "price": "MARKET"}
```

```
{"order_id": "7a32b14f-007b-467f-a626-3b3a00f8cb0f", "timestamp": 1752247607251, "symbol": "BTCUSDT", "side": "SELL", "type": "LIMIT", "quantity": 0.01, "price": 30000.0, "status": "NEW"}
```

```
{"oco_id": "67f1ffce-2194-4131-bf5a-31b13e1397a6", "timestamp": 1752334401699, "symbol": "BTCUSDT", "side": "SELL", "type": "OCO", "quantity": 0.01, "take_profit_price": 30000.0, "stop_price": 25900.0, "stop_limit_price": 39480.0, "status": "NEW"}
```

```
{"twap_id": "6e76afb7-8ba3-4043-951c-a0d21b1d624f", "slice": 1, "timestamp": 1752334429963, "symbol": "BTCUSDT", "side": "BUY", "type": "TWAP", "quantity": 0.008, "status": "FILLED"}
```

```
{"twap_id": "6e76afb7-8ba3-4043-951c-a0d21b1d624f", "slice": 2, "timestamp": 1752334431965, "symbol": "BTCUSDT", "side": "BUY", "type": "TWAP", "quantity": 0.008, "status": "FILLED"}
```

```
{"twap_id": "6e76afb7-8ba3-4043-951c-a0d21b1d624f", "slice": 3, "timestamp": 1752334433966, "symbol": "BTCUSDT", "side": "BUY", "type": "TWAP", "quantity": 0.008, "status": "FILLED"}
```

```
{"twap_id": "6e76afb7-8ba3-4043-951c-a0d21b1d624f", "slice": 4, "timestamp": 1752334435967, "symbol": "BTCUSDT", "side": "BUY", "type": "TWAP", "quantity": 0.008, "status": "FILLED"}
```

```
{"twap_id": "6e76afb7-8ba3-4043-951c-a0d21b1d624f", "slice": 5, "timestamp": 1752334437968, "symbol": "BTCUSDT", "side": "BUY", "type": "TWAP", "quantity": 0.008, "status": "FILLED"}
```

6. Future Enhancements

- Replace mock calls with real Binance Testnet API

- Integrate the Fear & Greed Index as a filter
 - Add Grid Strategy module
 - Build a Streamlit UI or simple Web dashboard
 - Backtest against provided historical data CSV
-

7. Screenshots

```

PROBLEMS  PORTS  OUTPUT  DEBUG CONSOLE  TERMINAL
powerShell

✓ MOCK Market Order Filled: {'order_id': 'f1798a7d-936c-4009-83db-0fad198e8ad7', 'timestamp': 1752247558132, 'symbol': 'BTCUSDT', 'side': 'BUY', 'type': 'MARKET', 'quantity': 0.01, 'status': 'FILLED', 'price': 'MARKET'}
PS C:\Users\Rutuja\.vscode\VsCodeProjects\ExecuTrade> python src/main.py limit BTCUSDT SELL 0.01 30000
✓ MOCK Limit Order Accepted: {'order_id': '7a32b14f-007b-467f-a626-3b3a00f8cb0f', 'timestamp': 1752247607251, 'symbol': 'BTCUSDT', 'side': 'SELL', 'type': 'LIMIT', 'quantity': 0.01, 'price': 30000.0, 'status': 'NEW'}
PS C:\Users\Rutuja\.vscode\VsCodeProjects\ExecuTrade>

```

```

PS C:\Users\Rutuja\.vscode\VsCodeProjects\ExecuTrade> python src/main.py oco BTCUSDT SELL 0.01 30000 25900 39480
✓ MOCK OCO Order Accepted: {'oco_id': '67f1ffce-2194-4131-bf5a-31b13e1397a6', 'timestamp': 1752334401699, 'symbol': 'BTCUSDT', 'side': 'SELL', 'type': 'OCO', 'quantity': 0.01, 'take_profit_price': 30000.0, 'stop_price': 25900.0, 'stop_limit_price': 39480.0, 'status': 'NEW'}
PS C:\Users\Rutuja\.vscode\VsCodeProjects\ExecuTrade> python src/main.py twap BTCUSDT BUY 0.04 5 2
✓ MOCK TWAP slice 1/5: {'twap_id': '6e76afb7-8ba3-4043-951c-a0d21b1d624f', 'slice': 1, 'timestamp': 1752334429963, 'symbol': 'BTCUSDT', 'side': 'BUY', 'type': 'TWAP', 'quantity': 0.008, 'status': 'FILLED'}
✓ MOCK TWAP slice 2/5: {'twap_id': '6e76afb7-8ba3-4043-951c-a0d21b1d624f', 'slice': 2, 'timestamp': 1752334431965, 'symbol': 'BTCUSDT', 'side': 'BUY', 'type': 'TWAP', 'quantity': 0.008, 'status': 'FILLED'}
✓ MOCK TWAP slice 3/5: {'twap_id': '6e76afb7-8ba3-4043-951c-a0d21b1d624f', 'slice': 3, 'timestamp': 1752334433966, 'symbol': 'BTCUSDT', 'side': 'BUY', 'type': 'TWAP', 'quantity': 0.008, 'status': 'FILLED'}
✓ MOCK TWAP slice 4/5: {'twap_id': '6e76afb7-8ba3-4043-951c-a0d21b1d624f', 'slice': 4, 'timestamp': 1752334435967, 'symbol': 'BTCUSDT', 'side': 'BUY', 'type': 'TWAP', 'quantity': 0.008, 'status': 'FILLED'}
✓ MOCK TWAP slice 5/5: {'twap_id': '6e76afb7-8ba3-4043-951c-a0d21b1d624f', 'slice': 5, 'timestamp': 1752334437968, 'symbol': 'BTCUSDT', 'side': 'BUY', 'type': 'TWAP', 'quantity': 0.008, 'status': 'FILLED'}
PS C:\Users\Rutuja\.vscode\VsCodeProjects\ExecuTrade>

```