1. Project Overview

This project is a command-line-based trading bot for Binance USDT-M Futures, developed using Python. It enables the execution of a wide range of order types, including both basic (market and limit) and advanced (stop-limit, OCO-simulated, TWAP, and grid) strategies.

The bot connects to Binance's **Futures Testnet API**, allowing full testing and validation of the trading logic without risking real funds. The primary objective of this project is to demonstrate skills in API integration, order placement, error handling, structured logging, and trading strategy simulation.

---

2. Features Implemented

2.1 Market Orders

Executes an immediate market buy or sell at the best available price.
Command Example:

```
python src/market_orders.py BTCUSDT BUY 0.01
```

2.2 Limit Orders

Places a limit order to buy or sell at a specific price, remaining open until filled or canceled.
Command Example:

```
python src/limit_orders.py BTCUSDT SELL 0.01 61000
```

2.3 Stop-Limit Orders

Triggers a limit order when the market reaches a predefined stop price.
Command Example:

```
python src/advanced/stop_limit.py BTCUSDT SELL 0.01 60000 59000
```

2.4 OCO (One Cancels the Other - Simulated)

Places both a take-profit limit and a stop-loss market order. Binance Futures does not support native OCO, so this is simulated manually.
Command Example:

```
python src/advanced/oco.py BTCUSDT BUY 0.01 61000 59000
```

2.5 TWAP (Time-Weighted Average Price)

Breaks a large order into smaller chunks and places each over a fixed interval.
Command Example:

```
python src/advanced/twap.py BTCUSDT BUY 0.05 5 10
```

2.6 Grid Strategy

Automatically places multiple buy or sell limit orders between a given price range.
Command Example:

```
python src/advanced/grid.py BTCUSDT BUY 0.01 59000 60000 5
```

3. Architecture and Design

- **CLI Interface: Every module utilizes argparse for command-line argument parsing.**
- **Environment Management: API keys are securely stored in a .env file and loaded with python-dotenv.**
- **Logging: All execution steps, errors, and order placements are logged in bot.log.**

Folder Structure:

```
binance_bot/
├── src/
│   ├── market_orders.py
│   ├── limit_orders.py
│   └── advanced/
│       ├── stop_limit.py
│       ├── oco.py
│       ├── twap.py
│       └── grid.py
├── bot.log
├── .env
├── requirements.txt
├── README.md
└── report.pdf
```

4. Sample Output Logs

Market Order (Success):

```
Success: Market BUY order placed for 0.01 BTCUSDT
```

OCO Validation Error:

```
Stop-Loss (60000.0) is too close to or below current price (119068.35).

Adjust it at least 50 USDT above.
```

TWAP Execution:

```
Chunk 1/5 placed: BUY 0.01 BTCUSDT
Chunk 2/5 placed: BUY 0.01 BTCUSDT
...
TWAP execution completed.
```

---

5. Challenges Faced

- **Stop Order Rejections: Initially, STOP_MARKET orders failed when the stop price was too close to the market price. This was solved by introducing a buffer of 50 USDT and validating prices against the mark price.**

- **Lack of Native OCO in Futures: Since Binance Futures does not support OCO natively, the feature was simulated using linked take-profit and stop-loss orders with manual cancellation.**

- **TWAP Timing and Splitting: Splitting quantity and adding delay between market orders required careful rounding and time.sleep management to ensure proper pacing.**

---

6. Learning Outcomes

- **Gained hands-on experience using Binance's Futures API with live data (testnet).**

- **Applied trading logic to real-world order types, including TWAP and grid.**

- **Learned secure API management using environment variables.**

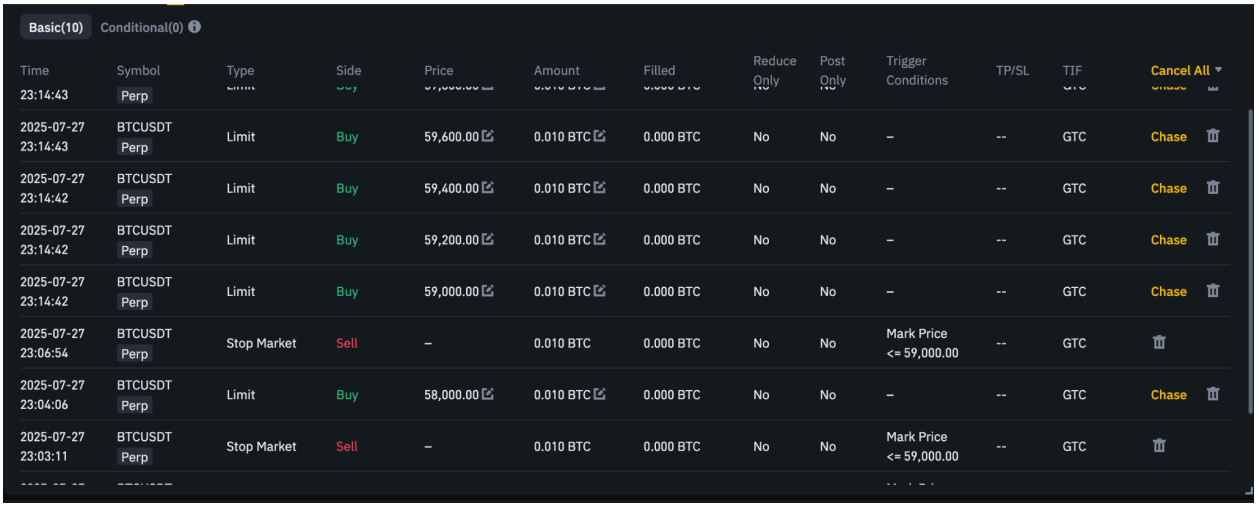
- **Improved Python skills in structured logging, input validation, and CLI tool development.**

---

7. Suggestions for Future Work

- **Automate OCO behavior using background monitoring and cancellation logic.**
- **Introduce trailing stop and dynamic take-profit features.**
- **Develop a graphical or web-based interface.**
- **Implement real-time order book analysis for smarter price placements.**
- **Add trade history tracking and PnL calculations.**

Binance Testnet – Open Orders Tab



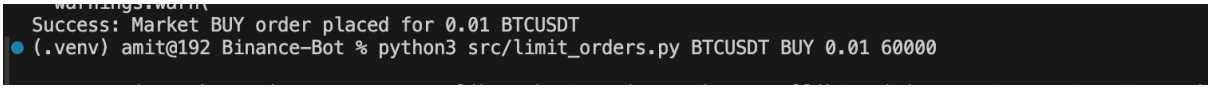
Basic(10) Conditional(0)

Time	Symbol	Type	Side	Price	Amount	Filled	Reduce Only	Post Only	Trigger Conditions	TP/SL	TIF	Cancel All
23:14:43	Perp											
2025-07-27 23:14:43	BTCUSDT Perp	Limit	Buy	59,600.00	0.010 BTC	0.000 BTC	No	No	–	--	GTC	Chase
2025-07-27 23:14:42	BTCUSDT Perp	Limit	Buy	59,400.00	0.010 BTC	0.000 BTC	No	No	–	--	GTC	Chase
2025-07-27 23:14:42	BTCUSDT Perp	Limit	Buy	59,200.00	0.010 BTC	0.000 BTC	No	No	–	--	GTC	Chase
2025-07-27 23:14:42	BTCUSDT Perp	Limit	Buy	59,000.00	0.010 BTC	0.000 BTC	No	No	–	--	GTC	Chase
2025-07-27 23:06:54	BTCUSDT Perp	Stop Market	Sell	–	0.010 BTC	0.000 BTC	No	No	Mark Price <= 59,000.00	--	GTC	
2025-07-27 23:04:06	BTCUSDT Perp	Limit	Buy	58,000.00	0.010 BTC	0.000 BTC	No	No	–	--	GTC	Chase
2025-07-27 23:03:11	BTCUSDT Perp	Stop Market	Sell	–	0.010 BTC	0.000 BTC	No	No	Mark Price <= 59,000.00	--	GTC	

Terminal Outputs -

A. Market Order (Success)

python src/market_orders.py BTCUSDT BUY 0.01



B. Limit Order

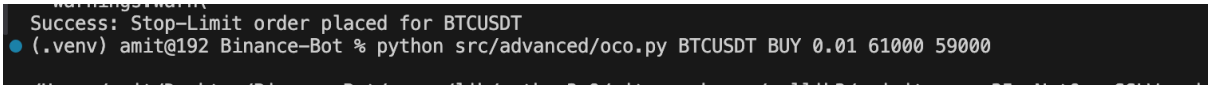
python src/limit_orders.py BTCUSDT SELL 0.01 61000

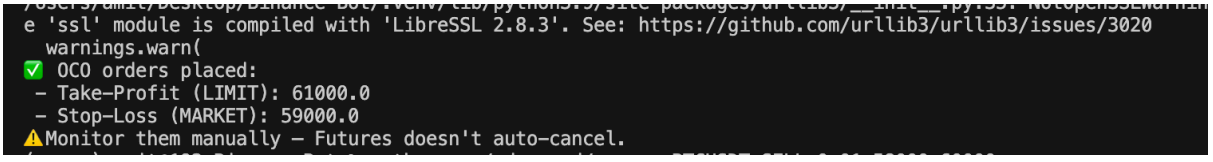
C. Stop-Limit Order

python src/advanced/stop_limit.py BTCUSDT SELL 0.01 60000 59000

D. OCO (Simulated)

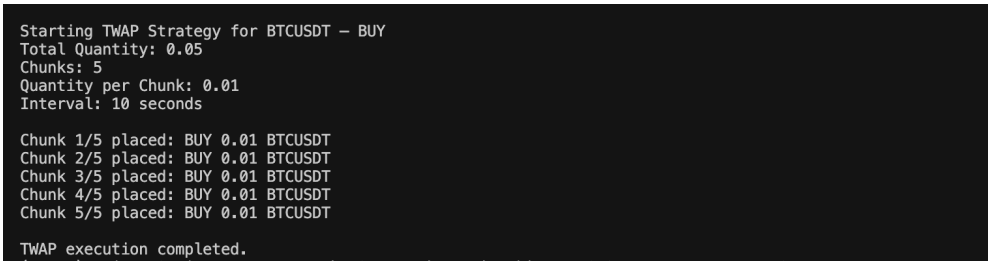
python src/advanced/oco.py BTCUSDT BUY 0.01 61000 59000





E. TWAP Execution

python src/advanced/twap.py BTCUSDT BUY 0.05 5 10



F. Grid Strategy

```
python src/advanced/grid_strategy.py BTCUSDT BUY 0.01 59000 60000 5
```

```
Placing Grid Orders for BTCUSDT - Side: BUY
Grid Levels: 5, Price Range: 59000.0 to 60000.0
Quantity per order: 0.01

Order 1/5 placed: BUY 0.01 BTCUSDT at 59000.0
Order 2/5 placed: BUY 0.01 BTCUSDT at 59200.0
Order 3/5 placed: BUY 0.01 BTCUSDT at 59400.0
Order 4/5 placed: BUY 0.01 BTCUSDT at 59600.0
Order 5/5 placed: BUY 0.01 BTCUSDT at 59800.0
```

/Bot.log

```
2025-07-27 22:17:40,933 - INFO - Placing market buy order for BTCUSDT
2025-07-27 22:17:40,933 - ERROR - Invalid quantity: must be greater than 0
2025-07-27 22:24:04,498 - ERROR - Binance API Error: APIError(code=0): Invalid JSON error message from Binance: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional
<HTML><HEAD><META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<TITLE>ERROR: The request could not be satisfied</TITLE>
</HEAD><BODY>
<H1>403 ERROR</H1>
<H2>The request could not be satisfied.</H2>
<HR noshade size="1px">
Request blocked.
We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or web
<BR clear="all">
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.
<BR clear="all">
<HR noshade size="1px">
<PRE>
Generated by cloudfront (CloudFront)
Request ID: Tqd-c9SFd0_Ap71e6_4-_Jf2J2QQoysCSEJ5feVSkT5zWYoe-UiRvA==
</PRE>
<ADDRESS>
</ADDRESS>
</BODY></HTML>
2025-07-27 22:35:05,532 - ERROR - Binance API Error: APIError(code=0): Invalid JSON error message from Binance: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional
2025-07-27 22:35:27,828 - ERROR - General Error: API Secret required for private endpoints
2025-07-27 22:40:50,744 - ERROR - General Error: API Secret required for private endpoints
2025-07-27 22:46:53,676 - ERROR - General Error: API Secret required for private endpoints
2025-07-27 22:49:42,685 - ERROR - Binance API Error: APIError(code=0): Invalid JSON error message from Binance: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional
2025-07-27 22:50:24,713 - INFO - Market order placed: BUY 0.01 BTCUSDT
2025-07-27 22:55:59,048 - INFO - Limit order placed: BUY 0.01 BTCUSDT at 60000
2025-07-27 22:57:37,888 - ERROR - Binance API error: APIError(code=-1106): Parameter 'price' sent when not required.
2025-07-27 22:59:50,355 - INFO - STOP (stop-limit) order placed: SELL 0.01 BTCUSDT @ 60000.0 (stop 59000.0)
2025-07-27 23:03:11,371 - INFO - OCO simulated: TP @ 61000.0, SL @ 59000.0, side: SELL, qty: 0.01
2025-07-27 23:04:06,185 - ERROR - Binance API error: APIError(code=-2021): Order would immediately trigger.
2025-07-27 23:06:54,497 - INFO - OCO Simulated — TP: 61000.0, SL: 59000.0, Side: SELL
2025-07-27 23:10:28,061 - INFO - TWAP chunk 1/5: BUY 0.01 BTCUSDT
2025-07-27 23:10:38,237 - INFO - TWAP chunk 2/5: BUY 0.01 BTCUSDT
2025-07-27 23:10:48,412 - INFO - TWAP chunk 3/5: BUY 0.01 BTCUSDT
2025-07-27 23:10:58,897 - INFO - TWAP chunk 4/5: BUY 0.01 BTCUSDT
2025-07-27 23:11:09,383 - INFO - TWAP chunk 5/5: BUY 0.01 BTCUSDT
2025-07-27 23:14:42,294 - INFO - Grid Order 1/5: BUY 0.01 BTCUSDT at 59000.0
2025-07-27 23:14:42,769 - INFO - Grid Order 2/5: BUY 0.01 BTCUSDT at 59200.0
2025-07-27 23:14:42,938 - INFO - Grid Order 3/5: BUY 0.01 BTCUSDT at 59400.0
2025-07-27 23:14:43,106 - INFO - Grid Order 4/5: BUY 0.01 BTCUSDT at 59600.0
2025-07-27 23:14:43,598 - INFO - Grid Order 5/5: BUY 0.01 BTCUSDT at 59800.0
```