Introduction

This project implements a Python-based trading bot that allows automated placement of Market, Limit, and Advanced orders on the Binance USDT-M Futures Testnet. The bot operates through simple CLI scripts and demonstrates secure API management, modular code design, and multiple order strategies.

Objectives

Place Market and Limit orders using the Binance Testnet

Store sensitive API credentials securely via .env

Implement logging and CLI-friendly execution

Extend functionality with advanced order types:

Simulated OCO (TP + SL)

TWAP Execution (Time-Weighted Average Price)

Grid Orders (spread-based)

Technologies Used

Python 3.8+

python-binance (Binance API SDK)

python-dotenv (for .env management)

PyCharm (IDE)

Binance Futures Testnet

Terminal/CLI for interaction

Project Folder Structure

bash

CopyEdit

```
sudhanshu_binance_bot/
├── .env              # API credentials (Testnet)
├── bot.log           # Order activity logs
├── report.pdf        # This document
├── README.md         # Setup and usage instructions
└── src/
    ├── market_orders.py    # Place market orders
    ├── limit_orders.py     # Place limit orders
    └── advanced/           # Advanced strategies
        ├── oco.py          # Simulated OCO (TP & SL)
        ├── twap.py         # Time-Weighted Avg. Price
        └── grid_orders.py  # Grid order placement
```

How It Works

Each script accepts parameters via CLI:

Market Order

bash

CopyEdit

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

Limit Order

bash

CopyEdit

```
python src/limit_orders.py BTCUSDT SELL 0.01 65000
```

Simulated OCO

You can import and call:

python

CopyEdit

```
place_simulated_oco("BTCUSDT", "BUY", 0.01, 67000, 63000)
```

TWAP Order

python

CopyEdit

```
twap_order("BTCUSDT", "BUY", 0.05, 5, 10)
```

Grid Order

python

CopyEdit

```
place_grid("BTCUSDT", "BUY", 65000, 100, 5, 0.01)
```

Logs and Error Handling

All scripts log important activity and errors into bot.log using Python's logging module.

Test Results

Script, Result

Market Orders, ✅Orders executed

Limit Orders, ✅Orders placed

OCO Strategy, ✅Simulated (TP + SL)

TWAP Execution, ✅Slices placed

Grid Strategy, ✅Multiple orders placed

Screenshots:

Tereminal:

Terminal   Local × + ⌄

✅Order 5/5 placed
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot>
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot> python src/advanced/grid_orders.py BTCUSDT BUY 65000 100 3 0.01
✅Grid order 1/3 placed at 64900.0
✅Grid order 2/3 placed at 64800.0
✅Grid order 3/3 placed at 64700.0
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot>
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot> python src/market_orders.py BTCUSDT BUY 0.01
✅Market order placed: BUY 0.01 BTCUSDT
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot> python src/limit_orders.py BTCUSDT SELL 0.01 65000
✅Limit order placed: SELL 0.01 BTCUSDT @ 65000.0
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot>

Bot log:

SB sudhanshu_binance_bot ⌄   Version control ⌄   Current File ⌄   ▷   🐞   ⋮   Trial

.env   limit_orders.py   bot.log ×   oco.py   twap.py   grid_orders.py   market_orders.py

4   2025-07-09 11:38:20,903 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
5   2025-07-09 11:38:42,649 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
6   2025-07-09 11:42:59,793 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
7   2025-07-09 11:46:13,826 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
8   2025-07-09 11:46:27,112 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
9   2025-07-09 11:54:05,419 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
10   2025-07-09 11:55:19,526 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
11   2025-07-09 11:56:18,197 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
12   2025-07-09 11:57:24,296 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
13   2025-07-09 12:00:18,155 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
14   2025-07-09 12:03:50,323 - ERROR - Market order error: APIError(code=-1021): Timestamp for this request was 1000ms ahead of the server's time.
15   2025-07-09 12:04:52,300 - INFO - Market Order: BUY 0.01 BTCUSDT | Order ID: 5229193381
16   2025-07-09 12:05:25,827 - INFO - Limit Order: SELL 0.01 BTCUSDT @ 65000.0 | Order ID: 5229194437
17   2025-07-09 13:11:11,507 - INFO - Market Order: BUY 0.01 BTCUSDT | Order ID: 5229371591
18   2025-07-09 13:11:12,887 - INFO - Limit Order: SELL 0.01 BTCUSDT @ 65000.0 | Order ID: 5229371625
19

Terminal   Local ×

✅Order 5/5 placed
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot>
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot> python src/advanced/grid_orders.py BTCUSDT BUY 65000 100 3 0.01
✅Grid order 1/3 placed at 64900.0
✅Grid order 2/3 placed at 64800.0
✅Grid order 3/3 placed at 64700.0
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot>
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot> python src/market_orders.py BTCUSDT BUY 0.01
✅Market order placed: BUY 0.01 BTCUSDT
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot> python src/limit_orders.py BTCUSDT SELL 0.01 65000
✅Limit order placed: SELL 0.01 BTCUSDT @ 65000.0
(.venv) PS C:\Users\LENOVO\Downloads\sudhanshu_binance_bot>

Binance Order History: