Binance Futures Trading Bot

Objective

The objective of this project is to develop a command-line based trading bot for Binance USDT-M Futures Testnet. The bot allows users to place market, limit, and advanced order types while ensuring proper validation, logging, and error handling.

System Overview

The trading bot is implemented in Python using the official Binance API via the python-binance library. It interacts exclusively with the Binance Futures Testnet environment to ensure safe testing without real funds.

Supported Features

- MarketOrders(Buy / Sell)
- Limit Orders (Buy / Sell)
- OCO (One-Cancels-the-Other) Orders
- Command-line interface for order execution
- Input validation and structured logging

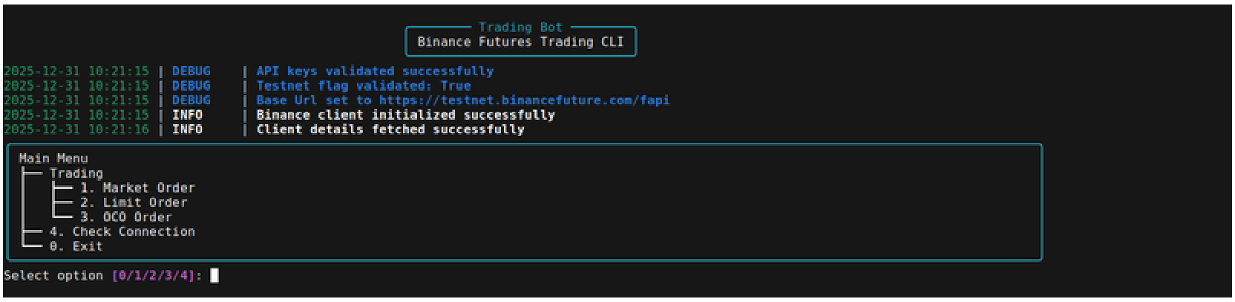
Project Structure

```
├── src/
│   ├── client.py             # Binance client (Testnet)
│   ├── market_order.p        # Market order logic
│   ├── limit_order.py        # Limit order logic
│   ├── utils/
│   │   ├── logging.py        # Logging configuration
│   │   ├── validators.py     # Input validation
│   │   └── __init__.py
│   ├── advanced/
│   │   ├── oco.py            # OCO order implementation
│   │   └── __init__.py
│   └── __init__.py
│
├── cli.py                    # CLI entry point
├── bot.log                   # Log file
├── .gitignore
└── README.md
```

Interactive Command line Interface

The trading bot comes with an easy-to-use interactive command-line interface that guides the user through different trading actions. When the application starts, it first checks the API keys, confirms the connection to the Binance Futures Testnet, and then displays a simple menu with available options.

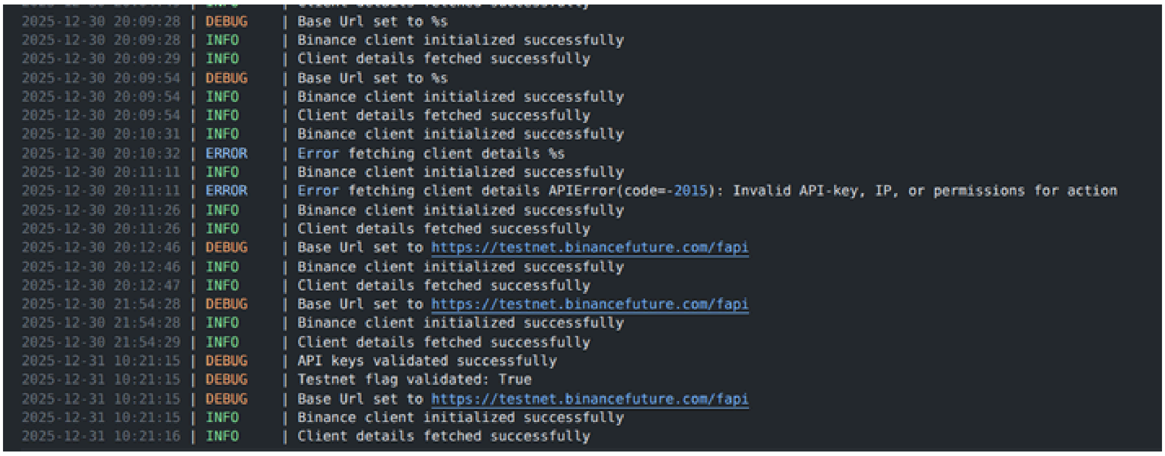
Users can place market orders, limit orders, OCO orders, or check the connection status directly from the menu. Clear prompts make it easy to enter inputs, and the bot immediately shows whether an order was placed successfully or if something went wrong.



```
                                       Trading Bot
                                Binance Futures Trading CLI
2025-12-31 10:21:15 | DEBUG    | API keys validated successfully
2025-12-31 10:21:15 | DEBUG    | Testnet flag validated: True
2025-12-31 10:21:15 | DEBUG    | Base Url set to https://testnet.binancefuture.com/fapi
2025-12-31 10:21:15 | INFO     | Binance client initialized successfully
2025-12-31 10:21:16 | INFO     | Client details fetched successfully
 Main Menu
 ├── Trading
 │   ├── 1. Market Order
 │   ├── 2. Limit Order
 │   └── 3. OCO Order
 ├── 4. Check Connection
 └── 0. Exit
Select option [0/1/2/3/4]: █
```

Logging & Error Handling

All API requests,responses, and errors are logged in a structured format inside the bot.log file. This ensures transparency, easier debugging, and traceability of all trading actions.



```
2025-12-30 20:09:28 | DEBUG    | Base Url set to %s
2025-12-30 20:09:28 | INFO     | Binance client initialized successfully
2025-12-30 20:09:29 | INFO     | Client details fetched successfully
2025-12-30 20:09:54 | DEBUG    | Base Url set to %s
2025-12-30 20:09:54 | INFO     | Binance client initialized successfully
2025-12-30 20:09:54 | INFO     | Client details fetched successfully
2025-12-30 20:10:31 | INFO     | Binance client initialized successfully
2025-12-30 20:10:32 | ERROR    | Error fetching client details %s
2025-12-30 20:11:11 | INFO     | Binance client initialized successfully
2025-12-30 20:11:11 | ERROR    | Error fetching client details APIError(code=-2015): Invalid API-key, IP, or permissions for action
2025-12-30 20:11:26 | INFO     | Binance client initialized successfully
2025-12-30 20:11:26 | INFO     | Client details fetched successfully
2025-12-30 20:12:46 | DEBUG    | Base Url set to https://testnet.binancefuture.com/fapi
2025-12-30 20:12:46 | INFO     | Binance client initialized successfully
2025-12-30 20:12:47 | INFO     | Client details fetched successfully
2025-12-30 21:54:28 | DEBUG    | Base Url set to https://testnet.binancefuture.com/fapi
2025-12-30 21:54:28 | INFO     | Binance client initialized successfully
2025-12-30 21:54:29 | INFO     | Client details fetched successfully
2025-12-31 10:21:15 | DEBUG    | API keys validated successfully
2025-12-31 10:21:15 | DEBUG    | Testnet flag validated: True
2025-12-31 10:21:15 | DEBUG    | Base Url set to https://testnet.binancefuture.com/fapi
2025-12-31 10:21:15 | INFO     | Binance client initialized successfully
2025-12-31 10:21:16 | INFO     | Client details fetched successfully
```

Conclusion

This project demonstrates clean architecture, proper API usage, and production-ready coding practices. It is designed to be easily extensible for future strategies such as Stop-Limit, TWAP, and Grid trading.